

图灵数学·统计学丛书32





有圣经之称的组合数学著作

Applied Combinatorics

应用组合数学

（第5版）

[美] Alan Tucker 著

冯速 译



应用组合数学（第5版）
Applied Combinatorics

“本书内容丰富，涉及面广，体系结构严整，简明易懂。”

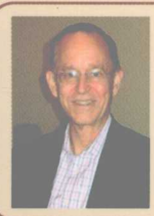
——《泰晤士高等教育增刊》

“这是我的大学教材，这么多年一直保留到现在，仍然需要不时查阅。强烈推荐给计算机和应用数学专业的学生，还有所有程序员。”

——Amazon 读者评论

组合数学是一门研究离散对象的科学，不仅在基础数学研究中具有重要的地位，在计算机科学、编码和密码学、物理、化学、生物等学科，以及企业管理、交通规划、战争指挥、金融分析等领域也都有重要的应用。

本书是组合数学领域的名著，有“圣经”之誉。自1980年初版以来不断更新和修订，至今依然畅销不衰。同时，它已被译为多种文字，被各国许多大学用作教材，产生了世界性的影响。书中主要讲述了离散数学问题求解中组合推理和建模的思想和方法，重视应用，详略得当，并通过贴近日常生活、容易理解的实际问题和游戏作为示例，大大增加了趣味性。



Alan Tucker 美国著名数学家和数学教育家。曾任美国数学协会（MAA）第一副主席。纽约州立大学石溪分校应用数学系教授，曾任斯坦福大学客座教授。1969年获斯坦福大学数学博士学位，师从线性规划之父Danzig。他出身数学世家，父亲和祖父都曾担任美国数学协会的主席。父亲Albert Tucker也是著名数学家，提出了囚徒困境和Kuhn-Tucker条件，培养了纳什和明斯基等大家。





www.wiley.com

本书相关信息请访问：图灵网站 http://www.turingbook.com

读者热线：(010)88593802

反馈/投稿/推荐信箱：contact@turingbook.com

分类建议 应用数学/组合学

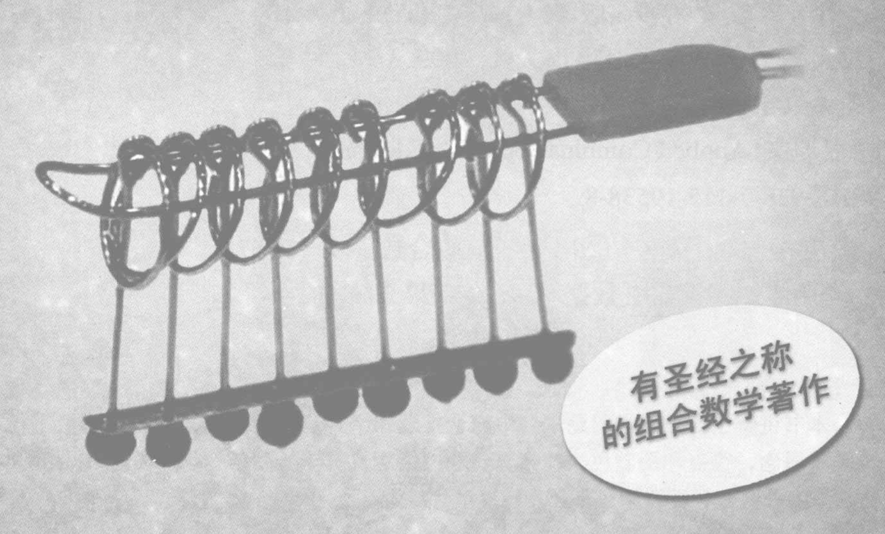
人民邮电出版社网址 www.ptpress.com.cn


ISBN 978-7-115-19538-8

ISBN 978-7-115-19538-8/O1

定价：65.00 元

TURING 图灵数学·统计学丛书 32



有圣经之称的组合数学著作



Applied Combinatorics

应用组合数学

（第 5 版）

[美] Alan Tucker 著

冯速 译

人民邮电出版社

北京

图书在版编目(CIP)数据

应用组合数学：第 5 版 / (美) 塔克 (Tucker, A.) 著；冯速译. —北京：人民邮电出版社, 2009. 3

(图灵数学·统计学丛书)

书名原文: Applied Combinatorics, 5th Edition

ISBN 978-7-115-19538-8

Ⅰ. 应··· Ⅱ. ①塔··· ②冯··· Ⅲ. 组合数学 Ⅳ. O157

中国版本图书馆 CIP 数据核字(2008)第 203458 号

内 容 提 要

本书讲解了离散数学问题求解中组合推理和组合建模的方法、思维和运用. 主要涉及图论基本概念、覆盖和图着色、搜索算法和网络运算算法等图论知识和方法, 以及基本的计数方法、生成函数计数模型、递推关系模型、容斥原理、Polya 枚举公式等枚举方法及其应用. 作者还介绍了如何用计算机科学地处理枚举, 以及逐步受限游戏的理论及其在尼姆游戏中的应用, 体现了组合数学的趣味性.

本书内容丰富, 简明易懂, 适合作为高等院校数学专业和计算机专业高年级本科生及研究生的教材, 也可供对组合数学有兴趣的相关人员阅读.

图灵数学·统计学丛书

应用组合数学（第 5 版）

◆ 著　[美] Alan Tucker
译　冯　速
责任编辑　明永玲
执行编辑　张继发

◆ 人民邮电出版社出版发行　北京市崇文区夕照寺街 14 号
邮编 100061　电子函件 315@ptpress.com.cn
网址：http://www.ptpress.com.cn
北京铭成印刷有限公司印刷

◆ 开本：700×1000　1/16
印张：22.75
字数：583 千字　2009 年 3 月第 1 版
印数：1–3 000 册　2009 年 3 月北京第 1 次印刷

著作权合同登记号　图字：01-2006-7685 号

ISBN 978-7-115-19538-8/O1

定价：65.00 元

读者服务热线：(010)88593802　印装质量热线：(010) 67129223

反盗版热线：(010)67171154

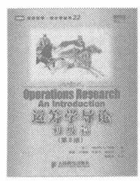


精选大师名作 打造专业品牌



图灵数学统计学系列丛书（含影印版和中文版）致力于引进国外优秀数学统计学教材。入选图书或为世界知名院校竞相采用的顶级教材，久为国内学界瞩目；或属于计算数学和金融数学等教学、科研和应用的热点领域，填补了国内教材的空缺；更有众多在国际学术界享有盛誉的大师名作，有些著作原版已经印行达40次以上，至今仍畅销不衰。丛书从选题、引进、翻译到出版，得到了齐民友、路见可、胡迪鹤、叶中行、陆金甫、龚光鲁等国内数学统计学界老中青三代专家的鼎力支持。

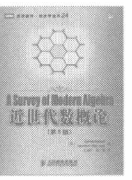
图灵原版数学·统计学系列 图灵数学·统计学丛书 高等院校数学·统计学教材系列



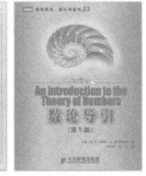

书　名：运筹学导论：初级篇（第8版）（中英文版）
原书名：Operations Research: An Introduction
作　者：[美] Hamdy A. Taha
译　者：薛毅 刘德刚 朱建明 侯思祥
中文版书号：978-7-115-18150-3
英文版书号：7-115-16066-9
中文版定价：69.00元
英文版定价：59.00元




书　名：运筹学导论：高级篇（第8版）（中英文版）
原书名：Operations Research: An Introduction
作　者：[美] Hamdy A.Taha
译　者：刘德刚 薛毅 朱建明 侯思祥
中文版书号：978-7-115-18947-9
英文版书号：7-115-16065-2
中文版定价：59.00元
英文版定价：59.00元

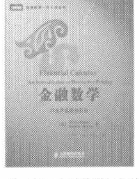
书　名：近世代数概论（第5版）（中英文版）
原书名：A Survey of Modern Algebra
作　者：[美] Garrett Birkhoff, Sauders Mac Lane
译　者：王连祥 徐广善
中文版书号：978-7-115-18387-3
英文版书号：7-115-16231-1
中文版定价：69.00元
英文版定价：69.00元

书　名：数论导引（第5版）（中英文版）
原书名：An Introduction to the Theory of Numbers, 5E
作　者：G.H.Hardy, Edward Maitland Wright
译　者：张明尧 张凡
中文版书号：978-7-115-18452-8
英文版书号：7-115-15611-2
中文版定价：69.00元
英文版定价：69.00元



书　名：统计模拟（第4版）（中英文版）
原书名：Simulation, 4E
作　者：[美] Sheldon M. Ross
译　者：王兆军 陈广雷 邹长亮
中文版书号：7-115-16085-0
英文版书号：7-115-15564-1
中文版定价：39.00元
英文版定价：45.00元




书　名：金融数学：衍生产品定价引论（中英文版）
原书名：Financial Calculus
作　者：Martin Baxter, Andrew Rennie
译　者：叶中行 王桂兰 林建忠（上海交通大学）
中文版书号：7-115-14204-1
英文版书号：7-115-14089-8
中文版定价：29.00元
英文版定价：35.00元




书　名：应用随机过程：概率模型导论（第9版）（中英文版）
原书名：Introduction to Probability Models
作　者：[美] Sheldon M. Ross
译　者：龚光鲁（清华大学）
中文版书号：7-115-16733-0
英文版书号：7-115-16023-2
中文版定价：89.00元
英文版定价：89.00元




书　名：随机过程初级教程（第2版）（中英文版）
原书名：A First Course in Stochastic Processes
作　者：Samuel Karlin, Howard M.Taylor
译　者：庄兴无 陈宗洵 陈庆华
中文版书号：7-115-16337-0
英文版书号：7-115-16598-5
中文版定价：69.00元
英文版定价：69.00元



书　名：金融数学教程（中英文版）
原书名：A Course in Financial Calculus
作　者：Alison Etheridge
译　者：张寄洲 等
中文版书号：7-115-14892-9
英文版书号：7-115-14090-1
中文版定价：29.00元
英文版定价：29.00元

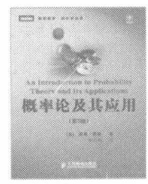



书　名：概率论基础教程（第7版）（中英文版）
原书名：A First Course in Probability, 7E
作　者：Sheldon M.Ross
译　者：郑忠国 詹从赞
中文版书号：7-115-15404-X
英文版书号：7-115-16541-1
中文版定价：49.00元
英文版定价：69.00元

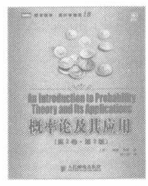


书　名：偏微分方程数值解（第2版）（中英文版）
原书名：Numerical Solution of PDE
作　者：K. W. Morton, D. F. Mayers
译　者：李治平 门大力等（北京大学）
中文版书号：7-115-14203-3
英文版书号：7-115-14075-8
中文版定价：29.00元
英文版定价：39.00元

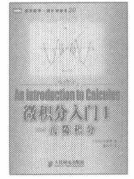
书　名：图论导引（中英文版）
原书名：Introduction to Graph Theory
作　者：Gary Chartrand, Ping Zhang
译　者：范益政 汪毅 龚世才 朱明
中文版书号：7-115-16153-6
英文版书号：7-115-14834-1
中文版定价：49.00元
英文版定价：49.00元



书　名：概率论及其应用
原书名：An Introduction to Probability Theory and Its Applications
作　者：William Feller
译　者：胡迪鹤（武汉大学）
书　号：7-115-14729-9
定　价：59.00元



书　名：概率论及其应用（第2卷）
原书名：An Introduction to Probability Theory and Its Applications
作　者：[美] William Feller
译　者：郑元禄
书　号：7-115-16735-4
估　价：79.00元



书　名：微积分入门I：一元微积分
原书名：解析入门I
作　者：小平邦彦(Kunihiko Kodaira)
译　者：裴东河
书　号：978-7-115-17261-7
定　价：39.00元



书　名：微积分入门II：多元微积分
原书名：解析入门II
作　者：小平邦彦(Kunihiko Kodaira)
译　者：裴东河
书　号：978-7-115-18373-6
定　价：39.00元



书　名：流形上的微积分（双语版）
原书名：Calculus on Manifolds
作　者：Michael Spivak
译　者：齐民友 路见可（武汉大学）
书　号：7-115-14224-6
定　价：29.00元
出版时间：2006年2月



书名：从微分观点看拓扑（双语版）
原书名：Topology from the Differentiable Viewpoint
作者：John Willard Milnor
译者：熊金城
书号：978-7-115-18467-2
定价：20.00元

邮　编：100097　地　址：北京市海淀区西四环北路140号京鼎原商务楼405室　人民邮电出版社 图灵公司市场部
电　话：010-88592341，88592340，88593802　传　真：010-88593803　E-mail：contact@turingbook.com　网址：http://www.turingbook.com

版 权 声 明

译　者　序

组合数学是系统分析的基础, 在计算机科学、数学、运筹学、生物学等诸多学科中有着广泛的应用. 组合数学内容繁多, 所要处理的问题又多是非常深奥难解的 NP 问题, 掌握组合数学的相当知识已很不易, 而熟练运用组合数学解决实际问题则更是困难. 通过一两本书就想要学会组合数学是不太可能的. 读者可以根据自己的需求, 选择合适的书籍, 循序渐进地学习组合数学的知识.

本书讲授离散数学问题求解中组合推理和组合建模的方法、思维和运用, 它强调组合推理的 3 个主要方面 —— 可能性的系统分析, 问题逻辑结构的探究, 以及精巧、灵活的设计, 从而有助于培养基础离散数学问题求解的能力.

本书有如下三大特色. 首先, 较多地使用了贴近日常生活、比较容易理解的游戏作为示例. 第二, 本书详略得当, 强调推理技巧: 只有在需要运用证明中的推理去解决应用问题时, 才给出证明, 否则不给出证明, 而只给出结果, 然后把这些结果运用于问题求解. 第三, 本书以组合数学在计算机科学、运筹学以及有限概率中的重要应用作为研究动力, 同时使用纸牌游戏中的概率或推理游戏等作为更有趣的学习背景. 这样可以使学习趣味性更强, 而且可以使读者免于陷入特殊应用的复杂场景中. 因此, 这是一本面向应用的组合数学教学参考书, 读者可以相对轻松、快速地学习组合数学的相关概念、技巧及其应用. 对理论学习感兴趣的读者也可以通过本书领略到组合数学的应用能力, 理解各概念和技术的用途, 为深入进行理论学习打下良好的基础.

译者对本书的翻译目标是易懂、通畅. 我们为此付出了巨大的努力, 感谢我爱人在本书翻译过程中所给予的支持和帮助. 但是, 由于译者能力有限, 难免有对原书理解不够的地方, 不周之处, 敬请读者指正.

译　　者

2008 年 10 月

冯速　1961 年沈阳出生, 1997 年获名古屋大学工学博士学位, 北京师范大学教授. 从教十余年, 从事计算机基础理论和计算机辅助教学的研究及计算机理论课程的教学工作. 对计算机教育有浓厚的兴趣. 曾译《编译器工程》、《数据结构: 从应用到实现》等多部教学参考书.

前　言

组合推理是所有计算机系统分析的基础. 它在离散的运筹问题及有限概率中扮演类似的角色. 计算机科学的两个最基本的数学问题是程序的速度及其逻辑结构. 速度涉及枚举程序执行每一步的次数, 而逻辑结构涉及流程图的形式. 为了优化有效生产或垃圾回收 (garbage collection), 对运筹算法的速度以及逻辑结构的分析需要组合数学. 确定等可能出现的某一特定子集中某个元素出现的概率时, 需要对这个子集的元素个数进行计数. 这样的组合概率是很多非参数统计试验的基础. 因此, 枚举和图论广泛、深入地运用于数学科学.

本书教数学专业的学生如何进行组合式推理和建模. 如同一本微积分教材培养读者分析问题的基本求解能力一样, 本书也非常注重培养离散数学问题的基本求解能力.

本书所强调的组合推理的 3 个主要方面包括: 可能性的系统分析, 问题逻辑结构的探究 (例如, 寻找可以着手的子情况, 或者首先对 3 个对象而不是 n 个对象来解决问题), 以及精巧、灵活的设计. 尽管我们提及组合数学在计算机科学、运筹学以及有限概率中的重要应用, 但这些应用大多用于激发研究的兴趣. 许多涉及这些概念的例子使用了更有趣的背景, 诸如纸牌游戏或推理游戏.

我们总是首先以例子作为理论的背景, 而只有在需要运用证明中的推理去解决应用问题时, 才给出证明. 否则, 我们不加证明地只给出结果, 例如各种递推关系的求解形式就是如此, 然后我们把这些结果运用于问题求解. 有时候, 我们陈述若干定理来使学生领略特定领域的理论风采.

几十年来, 美国数学协会推荐的大学课程一直把组合问题求解作为数学专业训练的重要组成部分. 在计算机专业课程中, 组合问题求解是诸多重要科目的基础. 实际上, 我们期望大部分使用本书的学生是计算机专业的学生. 不论是数学专业还是计算机专业, 作者相信理解这里所强调的推理技巧比掌握各种定义和实用技术更重要.

本书是为已经具备各方面能力的高年级学生和研究生设计的. 学生的能力越强, 布置的练习就可以越难. 根据学习内容的多少, 本书可以用于一学期或两学期[①]的课程. 它还可以作为一学期的应用图论课程, 或者一学期或半学年的枚举组合课程 (从第 5 章开始). 半学年的本科离散数学课程应该涵盖第 1~3 章和第

① 这里是指一学年分为四学期 (quarter) 的学制. —— 编者注

5~8 章的大部分内容, 如果时间允许, 再从其他章节选讲一些主题.

强烈建议主讲教师手头备一本与本书配套的教学参考手册. 这一手册广泛讨论了学生们对于特定主题的常见误解, 介绍了成功讲授这门课程的教学风格, 以及课程大纲样本 (每周作业、测试, 等等).

本书的第 5 版延续了前几版的特色. 例如, 第 5 版恢复了第 1 版开篇的 Mastermind 游戏. 也恢复了第 2 版关于形式语言和有限状态机的一章, 但采用了缩编形式. 第 2 版和第 3 版的部分枚举练习的参考答案也加进了本版. 本版还有一些全新的内容, 例如关于色多项式、运输问题以及 NP 完备问题. 当然, 还有一些新颖的练习. 在计数问题的有关章节中, 对原来的很多练习还作了数值上的改变 (以防学生直接套用以往的习题解答).

很多人对本书的初稿和第 1 版作出了很有用的评注, Jim Frauenthal 和 Doug West 的帮助特别大. 本书的想法可以追溯到 1969 年 George Dantzig 和 George Polya 在斯坦福大学讲授的组合数学课程, 当时我还是学习这门课程的一名学生. 使用过本书早前版本的很多教师向我提供了很多宝贵的反馈和建议, 我衷心希望这一切可以使这一版变得更好. 我要对他们表示真诚的感谢. 追本溯源, 是我的父亲 A. W. Tucker 引起我对组合数学及积极讲授这门课程的兴趣, 长期以来, 我父亲致力于确立有穷数学在数学中的作用, 并力主在本科数学教学中加大有穷数学的内容. 最后, 我还要特别感谢纽约州立大学石溪分校听我的组合数学课程的学生们. 正是他们教会我如何讲授这门课程.

Alan Tucker
纽约州立大学石溪分校

绪　论

本书旨在探索并开发组合推理方面的工具, 它在计算机科学和离散应用数学中是分析很多问题的基础. 作为这种推理的一个热身练习, 我们在这里介绍一种 Mastermind 游戏. 它是于 20 世纪 70 年代提出来的, 在英格兰得到广泛普及, 以至于在 1975 年举办了全国 Mastermind 锦标赛, 参赛者如潮. Mastermind 所使用的组合推理方法与构成本书基础的组合推理属于同一类型, 但它把这些推理运用于娱乐之中.

这种游戏的目标是猜测由涂有颜色的小木柱组成的密码. 这一密码是从红色 (R)、白色 (W)、黄色 (Y)、绿色 (G)、蓝色 (Bu) 和黑色 (Bk) 中选出的一行 4 个柱子. 给每一种猜测结果打分, 以此透露出这些猜测结果接近真实密码的程度. 而且, 选定密码的玩家还需要告知猜测者: (1) 在猜测结果中有多少代码柱其颜色及在行中的位置都正确; (2) 有多少代码柱颜色是正确的 (它在密码的某个地方出现), 但位置是错误的. 这两条信息分别以下面的形式记录: 信息 (1) 以黑键的形式记录, 而信息 (2) 则以白键的形式记录. 密码被正确地猜测出来时游戏结束, 即该猜测结果被给予 4 个黑键的评分. 例 1 中的猜测说明了这种评分过程的工作程序.

例 1　Mastermind 评分过程

密码 R Bu Y Y	柱　子	评　分	评　注
第 1 次猜测	Bu W G Y	●○	1 个代码柱至多只能得到 1 个键, 因此 Y 得到 1 个黑键
第 2 次猜测	Y G G Y	●○	
第 3 次猜测	Bk G G W		零分实际上非常有用, 因为它排除了所有 3 种颜色
第 4 次猜测	Bu Bu R R	●○	在密码中只有 1 个蓝色柱子, 所以只有 1 个位置也正确的蓝色柱子得到一个黑键

这种游戏与计数问题有密切的关系. 有多少种可能的密码? 有多少种黑白评分键的不同集合? 已知第一次猜测结果和它的评分后, 还有多少种可能的密码? 这一游戏在概念上与信息分类和检索中的若干关键问题类似. 在计算机对人类语言、化学混合物或其他复杂数据集的识别中, 我们必须对数据提出若干预设问题. 编译器通常以二分搜索树的形式执行一系列顺序检测, 以便识别文本中它们正在编译的命令. 很多与高效识别相关的理论分析需要用到与 Mastermind 游戏的计数问题相同的推理和技术.

尽管我们把 Mastermind 游戏的讨论与计数问题关联在一起, 但是这一游戏本身具有娱乐性, 因此我们鼓励读者与朋友一起做一做这个游戏. 在没有娱乐伙伴的情况下, 本节的练习给出了一些游戏, 具有足以确定密码的猜测次数和评分. 下面

的例子说明如何分析这些练习.

例 2 寻找密码

依据下面的猜测和评分, 在阅读下面的分析之前读者应该尝试着自己确定密码.

	柱 子	评 分
第 1 次猜测	W G Bu R	●○
第 2 次猜测	Bk Y R Bu	●○
第 3 次猜测	R R G Y	○○○
第 4 次猜测	Y R R W	●●

关键的颜色是红色. 第 3 次猜测有 2 个红色且有 3 分记分键, 表明密码中至少有 1 个红色柱子; 但因为键是白色的, 所以红色一定在位置 3 或位置 4. 然而, 猜测 4 的得分只有黑键, 而红柱子在位置 2 和位置 3; 因为猜测 3 表明位置 2 不可能是红色, 所以可以推断密码在位置 3 有 1 个红色柱子, 而且没有其他红色柱子 (否则在位置 2 的红色柱子将得到 1 个白键). 注意, 猜测 2 中的黑色键也一定属于位置 3 的红色柱子.

现在, 猜测 3 表明绿色和黄色柱子必定得到 3 个白键中的 2 个 (因为有 1 个红色柱子没有得到任何键). 因为黄色在密码中的某个位置上, 猜测 4 告诉我们黄色一定在位置 1. 所以密码有 Y_R_ 的形式. 此时已知猜测 2 和猜测 4 中的键是黄色和红色, 这两个猜测中的其他颜色 (黑色、蓝色和白色) 都不可能在密码中. 另外, 黄色也不可能在位置 2 和位置 4, 因为它出现在猜测 2 和猜测 3 的相应位置上却没有得到黑键. 因此密码剩下的唯一可能是 Y G R G. □

有关 Mastermind 游戏策略的更多信息, 可以在 Ault 的著作中找到, 见文献 [1].

练 习

1. 确定下面猜测的密码:

G	Y	R	Bu	○○○○
G	Bu	R	Y	●●○○

2. 对于下面的猜测, 确定两种可能的密码 (其中一种没有重复颜色):

Bk	Bu	Y	W	○○
R	W	Bu	G	●○○
Bu	G	R	G	○○
Y	R	W	Bu	●○○

3. 确定下面猜测的密码:

Bu	Bu	Bk	W	●○
Y	W	Bk	W	●●
W	Bu	R	G	●○
Y	Bu	Bk	R	●●

4. 确定下面猜测的两种可能密码:

R　G　Y　W　○○

Y　Bk　W　Bu　●○

G　R　R　G　●○

Bk　Bu　G　R　●○

W　R　Bk　R　○

5. 寻找第 4 次猜测, 它的得分可以使你确定下面猜测的密码:

G　Y　Bk　R　●●

Y　Bu　G　W　●○

Bu　W　Y　Y　○

6. 寻找第 4 次猜测, 它的得分可以使你确定下面猜测的密码:

G　R　Bu　W　●

G　Y　Bu　G　●

Y　Bk　Bk　Y　○

7. 把第 7 种颜色橙色 (O) 加到这一游戏中. 确定下面猜测的密码:

O　R　Bu　G　○○

Y　O　W　Y　○

R　Bk　Bu　G　○○○

O　R　R　Y　●○

Bk　Bu　O　R　○○

8. 把第 7 种颜色橙色 (O) 加进来. 确定下面猜测的 2 种可能密码:

O　R　G　Bu　○○

Y　Y　W　O　●○

R　Bu　Bu　Bk　●○

Bk　O　Y　R　●○

Bu　Bk　O　Y　●○○

9. 把橙色 (O) 加进来. 确定下面猜测的 7 种可能密码:

O　Y　W　Bk　○

W　Bk　O　R　●○

G　O　R　R　●

Y　R　Bu　O　○○

10. 现在, 有 5 个位置和 2 种其他颜色: 橙色 (O) 和粉色 (P). 确定下面猜测的密码:

Bk　G　G　O　O　●●○

W　O　Bk　G　R　●●○

O　Bk　P　Y　Bu　●○

G　P　Y　R　W　○○

11. 现在, 有 5 个位置和 3 种其他颜色: 橙色 (O)、粉色 (P) 和紫色 (V). 确定下面猜测的密码:

R　V　Bk　G　O　○○

V　R　V　R　Y　●○

Bk　W　W　Bu　Bk　●

P　O　P　O　G　○○

G Y Bu W P ●○○
W P V Bu Y ●○○
Y Bu R V Bk ○○○

12. 确定在 Mastermind 游戏中初始猜测的正确概率.

13. **(a)** 在 Mastermind 记分中, 不可能出现的 4 种或少于 4 种的黑白键集合是什么?
(b) 在 Mastermind 记分中, 有多少种不同的黑白键集合?

14. 称 Mastermind 中的两个猜测类似, 如果其中一个猜测可以通过对另一个猜测的位置置换或颜色置换而得到. 有多少不同 (非类似) 的猜测?

15. 假设你在第一次猜测使用了 4 种不同的颜色, 并且这一猜测的得分是 4 个白键. 有多少种不同的可能密码? (注意, 所有的 4 种颜色的猜测都是类似的, 参见练习 14.)

16. 假设你的第一次猜测使用了 3 种不同的颜色 (有一种颜色是重复的), 并且这一猜测的得分是 1 个黑键 3 个白键. 有多少种可能的密码? (注意, 所有的 3 种颜色的猜测都是类似的, 参见练习 14.)

17. 假设你的第一次猜测至少使用了 2 种颜色, 但它的得分是没有键. 这一猜测至少能排除多少种密码?

18. 考虑有 4 个柱子的简化 Mastermind 游戏, 每个柱子一种颜色, 而密码是由这 4 个柱子的某个排列组成的. 找出这一游戏的一个完备策略, 使得你能够在最多 3 次猜测之后确定密码 (在第 4 次猜测得到 4 个黑键的得分.)

19. 考虑有 3 个位置和 3 种柱子颜色的简化 Mastermind 游戏. 为这一游戏寻找最佳的第一次猜测. 称一次猜测是*最佳的*, 如果对于某个数 k, 这个猜测的所有可能得分最多剩下 k 种可能密码, 而其他任意猜测的某个可能得分至少剩下 k 种可能密码.

20. 考虑有 4 个位置但只用 2 种柱子颜色的简化 Mastermind 游戏. 寻找这一游戏的最佳第一次猜测 (参见练习 19).

21. 证明: 无论下面的 4 种猜测的得分是什么, 我们都可以 (使用前 4 次猜测) 在至多 2 次猜测之后正确地猜出任意的密码.

R R W W
R Bk R Bk
Bu Bu G G
Bu Y Bu Y

22. 编写一个构建密码并对猜测评分的计算机程序.

23. 编写一个 Mastermind 游戏的程序.

参考文献

1. L. Ault, *Mastermind Official Handbook*, Signet Press, New York, 1976.

目　　录

第一部分　图　　论

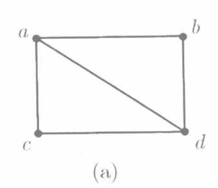


第1章 图论入门

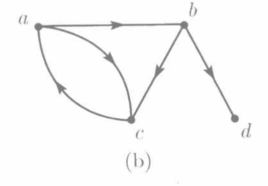
1.1 图模型

前 4 章研究图及其应用. 一个**图** (graph) $G=(V,E)$ 是由顶点 (vertex) 的有穷集合 V 和连接两个不同顶点的边 (edge) 的集合 E 组成的.[①] 图 1.1a 给出一个 $V=\{a,b,c,d\}$, $E=\{(a,b),(a,c),(a,d),(b,d),(c,d)\}$ 的图的描述. 我们用点表示顶点, 用连接一对顶点的线表示边. 图的定义不允许有两条连接相同顶点对的边. 另外, 边也不能 “循环”, 致使边的两个端点结束于相同的顶点, 即边的两个端顶点必须是不同的. 无向边的两个端点可以写成 (b,c) 或 (c,b) 的形式. 对于顶点 a 和 b, 当其间存在一条边 (a,b) 时, 我们说 a 和 b **相邻** (adjacent).

有时候, 边是有序的顶点对, 称这样的边为**有向边** (directed edge). 在**有向图** (directed graph) 中, 所有的边都是有向的. 参见图 1.1b 中的有向图. 我们用 $(b\vec{\ },c)$ 来表记从 b 到 c 的有向边. 在有向图中, 允许在一对顶点间的两个方向上各有一条边. 参见图 1.1b 中的边 $(a\vec{\ },c)$ 和 $(c\vec{\ },a)$.



(a)



(b)

图 1.1

图论中所需的组合推理, 以及本书后面的枚举部分中所需的组合推理, 所涉及的分析不同于数学分析或高中数学所使用的分析. 几乎没有解决这些问题的一般规则或公式. 每一种问题通常都需要根据情况作特殊分析. 有时候, 这样的分析需要构建巧妙的模型或需要有创造性的思想, 但更常见的是, 需要把问题分解成使用简单的逻辑或基本的计数原理就可以解决的多种情况 (及子情况). 推理的相关线索就是解决给定问题的特殊情况, 然后把这一推理推广到可能存在的所有情况. 在图论中, 可以使用图的直观性而使组合论证变得简单一些. 例如, 当你能够画出每一种情况的图示描述时, 那么分情况论证就很容易构造出来.

事实证明, 若一组元素中的各对元素因某种性质而相关, 则图是一种非常有用

① 本书中所说的图是大部分图论书中所谓的简单图 (simple graph). 一般来说, 图论这一术语在各书里有一些微小的不同.

的分析工具. 最明了的图的例子就是有物理链接的设备, 例如电子网络, 其中电子组件 (晶体管) 是顶点, 而连接晶体管的电线是边; 另一个例子就是电话通信系统, 其中电话和交换中心是顶点, 而电话线是边. 交通图、输油管道和地铁系统也都是图的例子.

图的另一种自然形式是有逻辑或继承顺序的集合, 例如, 计算机流程图, 其中指令是顶点, 而从一个指令到可能的后继指令的逻辑流是边; 又如组织结构图, 其中人是顶点, 而如果 A 是 B 的直接上级, 那么就有边 $(A\rightharpoonup, B)$. 计算机数据结构、生物中的进化树以及复杂项目中的任务调度等也都是这种图的例子.

本书的重点是对各种图及应用图模型的问题进行求解. 你会发现通常我们不与数值打交道, 而只与顶点和边打交道. 乍看起来, 这似乎完全是非数学的. 它确实与高中或微积分课程中所讲的数学完全不同. 然而, 诸如计算机科学和运筹学等学科既涉及标准的数值数学也涉及大量的图论.

本节包含若干图的说明性示例. 我们利用少量的逻辑和系统分析从零开始解决每一个问题. 在后续的章节中, 要对很多例子再作深入的研究.

下面 3 个图论术语要在后面的例子中使用. **路径** (path) P 是顶点的序列, 写作 $P = x_1 - x_2 - \cdots - x_n$, 其中相继的两个顶点由一条边相连. 如果还存在边 (x_n, x_1), 那么这个序列称为**回路** (circuit), 写作 $x_1 - x_2 - \cdots x_n - x_1$. 例如, 在图 1.1a 中, $b - d - a - c$ 形成一条路径, 而 $a - b - d - c - a$ 形成一条回路. 一个图是**连通的** (connected), 如果每一对顶点间都存在一条路径. 从一个连通图 G 中删除某些边或顶点, 使得结果图不再连通时, 则称删除这些边或顶点断开了 (disconnect) 这个图; 这时, 至少有一对顶点不再由一条路径相连. 图 1.1a 是连通的, 但是删除边 (a, b) 和 (b, d) 就会断开它.

例 1 匹配

假设有 5 个人 A, B, C, D, E 和 5 项工作 a, b, c, d, e, 不同的人胜任不同的工作. 问题是寻找一个人与工作的可行的一对一匹配, 或证明不存在这样的匹配. 可以用图表示这种情况, 其中, 用顶点代表每一个人和每一项工作, 而用边连接人与其所胜任的工作. 对于图 1.2, 存在这样的人与工作的可行匹配吗?

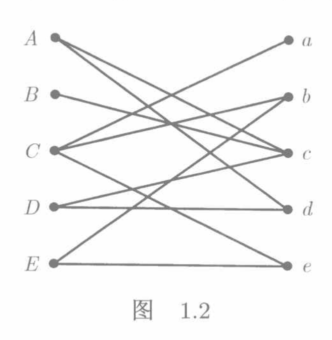


图 1.2

答案是否定的. 通过考虑人 A, B, D, 我们就可以发现其中的原因. 这 3 个人只能胜任两项工作 c 和 d. 因此对这 3 个人来说没有可行的匹配, 更何况共有 5 个人. 第 4 章将给出寻找可行匹配的算法, 当然是在可行匹配存在的前提下. 所有的边连接的两个顶点分属于两个集合的匹配图称为二部 (bipartite) 图. 我们将在 1.3 节中进一步讨论二部图. □

例 2 拼写检查器

拼写检查器检查文档中的每一个单词 X(在计算机它被表示为一个二进制数), 并尝试着把这个 X 与词典中的某个单词配对. 词典通常包含近 100 000 个单词. 为了弄明白这一检查是如何进行的, 考虑把一个未知的字母 X 与英语字母表中的 26 个字母中的一个进行匹配的简化问题. 按我们人类在词典中寻找给定单词出现的页码的方法, 计算机搜索程序首先把未知字母 X 与 M 比较, 以便确定是 X⩽M, 还是 X>M. 比较的结果确定了把 X 定位在字母表的前 13 个字母还是后 13 个字母中, 因此使可能是字母 X 的字母数量减半. 这种使可能匹配减半的策略一直持续下去, 直到找到这个字母 X 的位置. 例如, 如果 X⩽M, 那么可以检测是否有 X⩽G; 如果 X>M, 那么可以检查是否有 X⩽S.

可以用称为**树** (tree) 的有向图自然地表示这一检测过程. 图 1.3 给出这一字母匹配过程的前 3 轮比较. 顶点表示比较中所使用的字母. 顶点 Q 左边的下降边指向 X⩽Q 时下一次要比较的字母, 而 Q 右边的下降边指向 X>Q 时下一次要比较的字母.

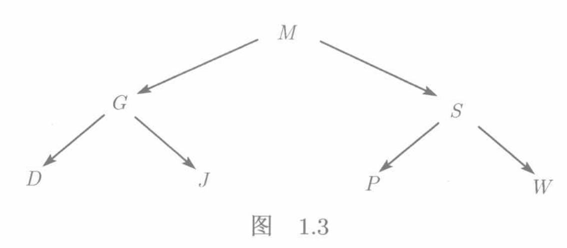


图 1.3

对于原来的拼写检查器问题, 文字处理软件将使用类似但更大的比较树. 仅用 12 轮比较就能够把与未知单词 X 可能匹配的单词数量从 100 000 减少到 25, 这个数量大致等于词典一页中的一列单词数目. (一旦减少到只有大约 25 种可能, 计算机通常线性地在这个列表中搜索 X, 就如我们人类所做的一样.) □

第 3 章研究树及其在各种搜索问题中的应用. 我们可以把树刻画成连通且任意一对顶点间有唯一一条路径 (忽略有向边的方向) 的图. 下面的例子按完全不同的方式使用树.

例 3 网络的可靠性

假设图 1.4 表示电话线 (或电线) 网络. 我们要研究这样的网络由于意外断开引发的不可靠性. 我们想要找出哪些线路和交换中心必须正常运作才能避免网络的断开.

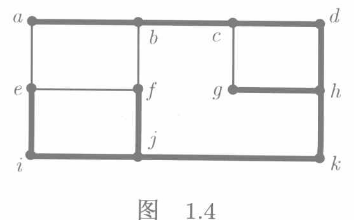


图 1.4

不存在这样的电话线 (边), 它的移除会使电话网络 (图) 断开. 同样, 也不存在这样的顶点, 它的移除会使网络断开.

是否存在某对边, 它的移除使这个图断开呢? 有若干对这样的边. 例如, 如果删除与 a 相连的两

条边, 则顶点 a 就从网络中孤立出来. 更有意思的断开边对是 $(b,c),(j,k)$. 试在图 1.4 中, 寻找出由两条边组成的所有断开的集合.

采用另一种行动方针. 假设我们要寻找把图 1.4 中的 11 个顶点链接到一起时所需的边的最小集合. 答案是存在若干边的最小连通集合. 通过观察, 我们发现下面的边的最小连通集合：(a,b)、(b,c)、(c,d)、(d,h)、(h,g)、(h,k)、(k,j)、(j,f)、(j,i) 和 (i,e). 在图 1.4 中, 黑线表示了这个最小连通集合中的边. 最小连通集合总是树. 关于这些集合有一个有趣的一般结果：如果图 G 有 n 个顶点, 那么 G 的最小连通集合 (如果存在的话) 总有 $n-1$ 条边. □

与一个顶点连接的边的数量称为这个顶点的**度** (degree).

例 4　街区监控

现在假设图 1.4 表示一个城市街道图的一部分. 我们想要在街角处安置警察, 使他们可以监控每一片街区 (边), 即, 每条边至少在其端点之一有一名警察. 那么能够完成这一工作的警察的最少数量是多少？

我们尝试着得到所需警察数量的下界. 这张图有 14 片街区 (边). 拐角 b、c、e、f、h 和 j 的度都为 3, 而拐角 a、d、g、i 和 k 的度都为 2. 因为 4 个顶点至多可以连接到 $4\times3=12$ 条边, 但是图中共有 14 条边, 所以我们至少需要 5 名警察. 现在, 我们尝试着寻找可以连接到所有边的 5 个顶点的集合. 如果能够找到这样的集合, 我们知道这是最理想 (最少) 的解决方案.

如果 5 名警察都被安置在度为 3 的顶点处, 那么 5 名警察可以监控 $5\times3=15$ 条边. 因为只有 14 条边, 所以某条边由两端顶点处的警察共同负责. 如果有 4 名警察位于度为 3 的顶点处, 而 1 名警察位于度数为 2 的顶点处, 那么正好可以监控 14 条边, 而且没有边需要两个端点同时得到监控. (如果 5 名警察中位于度为 3 的顶点不到 4 名警察, 那么就不能监控所有 14 条边.) 通过这些观察, 我们现在可以用系统分析的方法来寻找可以连接到所有 14 条边的 5 个顶点.

考虑边 (c,d). 假设它由顶点 d 处的警察监控, 那么顶点 c [边 (c,d) 的另一个端点] 不能有警察, 因为上面已经提到, 如果我们使用一个度为 2 的顶点, 比如说这里的 d, 那么没有边由两个端顶点同时监控. 然而, 如果不使用顶点 c, 那么边 (c,g) 必须由它的另一个端点 g 监控. 这样将会使用两个度为 2 的顶点 d 和 g. 上面我们提到最多只能使用一个度为 2 的顶点. 边 (c,d) 由顶点 d 监控的假设会使我们陷入困境.

现在, 我们假设在顶点 d 处没有警察, 那么必须让顶点 c 处的警察监控边 (c,d). 同样, 边 (d,h) 只能由位于顶点 h 处的警察监控. 下面来看边 (h,k). 它已经由顶点 h 监控, 于是我们认为 (h,k) 不能同时由顶点 k 处的警察监控, 因为 k 的度为 2, 而且我们在上面观察到：如果使用一个度为 2 的顶点, 那么没有边同时由两个端点监控. 我们得出在顶点 k 处不可能存在警察的结论. 于是边 (k,j) 只能由顶点 j 处的警察监控. 现在我们已经有了位于顶点 c, h, j 处的 3 名警察.

类似的推理表明, 因为有一名警察位于顶点 j, 所以在顶点 i 处没有警察, 那么必须有一名警察位于顶点 e, 于是在顶点 a 处不能有警察, 而必须有一名警察位于顶点 b. 综上所述, 我们已经证明应该在顶点 c、h、j、e 和 b 处安置警察. 简单的检查可以确认 5 个顶点的确可以监控所有 14 条边. 于是, 我们找到了这一监控问题所需的最少警察数量. 注意, 因为我们的推理迫使我们使用这 5 个顶点, 没有其他可行的五顶点集合.

在本例子开始时, 我们已经证明至少需要 5 个街角才能使所有街区 (边) 都在监控之下. 现在, 我们已经构建了这样的 5 个街角的集合来完成监控. 于是 5 是街角的最少数量.

最后, 对于这样的监控问题, 我们还可以考虑监控顶点而不是边: 至少需要多少警察来监控所有顶点? 我们使用与街区监控问题同样的推理, 来得到街角监控所需要的街角数量的下界. 顶点 x 处的警察监控顶点 x 及与 x 相邻的所有顶点. 一共有 11 个顶点, 其中有 6 个顶点各监控 4 个顶点 (它们自身和 3 个相邻的顶点). 因此, 理论上 3 是最小值. 事实上可以达到这样的最小值. 细节分析留作练习. □

图 G 中有如下性质的顶点集合 C 称为**边覆盖** (edge cover): G 的每条边至少与 C 的一个顶点相连. 前面的例子寻找图 1.4 中边覆盖的最小值. 例 4 中的推理展示了图论中常见的系统个案分析.

例 5　会议安排

考虑下面的调度问题. 州议会有很多委员会每周要举行为时 1 小时的会议. 有人想要做一份委员会会议时间安排表, 它能够使会议的总小时数最少, 而且使得有共同成员的两个委员会不在同一时间开会.

可以用下面这样的图来对这一情况建模: 在这个图中创建代表每一个委员会的顶点, 如果这两个顶点所表示的委员会有共同成员, 则用一条边连接这两个顶点. 现在假设图 1.4 表示 11 个立法委员会的会员重叠情况. 例如, 图 1.4 中顶点 c 到顶点 b、d 和 g 的边表示委员会 c 分别与委员会 b、d 和 g 有共同成员.

若干委员会可以在相同的时间举行会议, 如果这些委会所对应的顶点集合不存在边. 任意两个顶点都不存在边的顶点集合称为顶点的**独立集合** (independent set). 现在, 我们的会议安排问题可重述为: 寻找包含所有顶点的独立集合的最小数. 我们将在 2.3 节对这个问题进行更深层的讨论.

在某一时间有多少委员会可以举行会议呢? 我们是在提出这样的图问题: 这个图的最大独立集合是什么? 一般来说很难寻找图的最大独立集合. 对于图 1.4, 简单的检查就可以证明存在大小为 6 的独立集合: a、d、f、g、i 和 k. 所有其他独立集合都少于 5 个顶点.

图论的另一个目标就是在看似毫无关系、起源于不同背景的两个图概念间寻找有用关联. 现在, 我们证明独立集合与边覆盖有着密切的关系. 如果 V 是图 G 的顶点集合, 那么 I 是顶点的独立集合当且仅当 $V-I$ 是边覆盖! 为什么呢? 因为, 如

果在 I 中两个顶点间不存在边, 那么每条边 (至少) 包含一个不在 I 中的顶点, 即有一个顶点在 $V-I$ 中. 相反, 如果 C 是边覆盖, 使得所有边至少有一个端顶点在 C 中, 那么不存在连接 $V-C$ 中两个顶点的边. 所以 $V-C$ 是独立集合. 检查图 1.4 可知, 不在独立集合 a、d、f、g、i 和 k 中的顶点形成边覆盖 b、c、e、h 和 j.

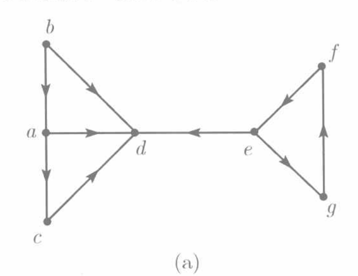
这种关系的一个推论是: 如果 I 是图 G 的最大可能的独立集合, 那么 $V-I$ 就是最小可能的边覆盖. 所以寻找最大的独立集合等价于寻找最小的边覆盖. □

下面我们给出一个涉及有向图的例子.

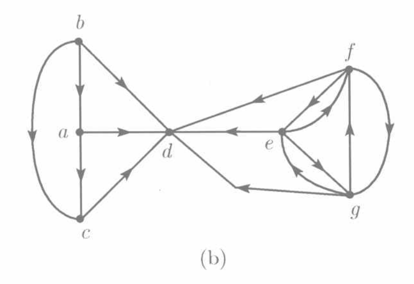
例 6 影响模式

假设对一组人进行心理学探测, 确定这组人中哪些成员会影响其他成员的想法. 我们可以创建这样一个图: 其顶点代表每一个人, 只要 p_1 影响 p_2 则画有向边 $(\overrightarrow{p_1, p_2})$. 设图 1.5a 表示这样一个影响集合. 现在, 我们寻找这样一些人的最小子集: 他们可以把一种想法传播给小组中的所有人, 或者直接影响他人, 或者对影响另一个人的人产生影响, 如此继续下去. 用图论的话来说, 此时我们要寻找到这样的顶点集合的最小子集: 对于其他所有顶点 x, 存在从该集合中某个顶点到 x 的有向路径 [从 p_1 到 p_k 的*有向路径* (directed path)是边序列 $(\overrightarrow{p_1, p_2}), (\overrightarrow{p_2, p_3}), \cdots, (\overrightarrow{p_{k-1}, p_k})$]. 影响顶点的这种子集称为**顶点基** (vertex basis).

为了求顶点基, 可以为原来的图构建一个*有向路径图* (directed-path graph), 其中顶点集合与原图相同, 如果在原图中存在从 p_i 到 p_j 的有向路径, 则画有向边 $(\overrightarrow{p_i, p_j})$. 图 1.5b 给出了图 1.5a 的有向路径图. 现在, 原来的问题可以重述为在新图中寻找这样的最小顶点集合: 对于所有其他顶点 x, 存在从该集合中顶点到 x 的有向边. 这正是例 4 结尾时所提到的顶点覆盖问题的有向图版本. 观察发现, 图 1.5b 中所有没有进入边的顶点必须在最小子集中 (因为没有其他顶点有到它的边), 顶点 b 就是这样的顶点. 因为 b 有到 a、c 和 d 的边, 所以 e、f 和 g 是剩下要受 "影响" 的顶点. e、f 和 g 之一 "影响" 这 3 个顶点. 所以 b、e 或者 b、f 或者 b、g 是要求的最小顶点子集. □



(a)



(b)

图 1.5

练 习 1.1

练习概况 前 6 道练习都是简单图模型. 练习 7~24 给出例子, 并扩展本节例子中所提到的

模型.

1. 假设有 5 条州际高速公路如下连接 6 个城镇 A、B、C、D、E 和 F: I-77 从 B 开始经由 A 到达 E; I-82 从 C 开始经由 D 然后再经由 B 到达 F; I-85 从 D 开始经由 A 到达 F; I-90 从 C 开始经由 E 到达 F; I-91 从 D 开始到达 E.

(a) 画出一个网络图, 其顶点代表城镇, 边代表连接相邻城镇的州际高速路.

(b) 删除某些边将阻止某两个城镇间的来往, 这样的边的最小数量是什么?

(c) 是否可能进行这样的一次从城镇 C 开始的旅行: 经过每一个城镇而每条边用的州际高速公路编号都各不相同 (这次旅行不必返回 C)?

2. (a) 假设有 4 支队 Aces、Birds、Cats 和 Dogs 相互进行比赛. Aces 打败了除 Birds 之外的所有其他对手. 除了赢了 Aces 之外, Birds 输给了其他所有对手. Dogs 打败了 Cats. 使用有向图表示这一游戏的结果.

(b) 优势顺序是各队的这样一个列表: 在这个顺序中的第 i 支队打败第 $(i+1)$ 支队. 寻找 (a) 中的所有优势顺序.

3. (a) 对 5 支足球队的比赛进行安排. 每支队要与另外两支队比赛. 解释如何创建这个问题的图模型.

(b) 证明: 除交换队名外, 问题 (a) 只存在一种可能的图.

4. 假设有 6 个人, 分别是 John、Mary、Rose、Steve、Ted 和 Wendy, 他们之间口头传话. 每天, John 都与 Mary 和 Wendy 交流; Mary 与 John、Rose 和 Steve 交流; Rose 与 Mary、Steve 和 Ted 交流; Steve 与 Mary、Rose、Ted 和 Wendy 交流; Ted 与 Rose、Steve 和 Wendy 交流; 而 Wendy 与 John、Steve、Ted 交流. 这些人把一天听到的无论什么都在第 2 天传给其他人.

(a) 利用图对这一传话状况建模.

(b) 把一个传言从 John 传到 Steve 需要多少天? 谁将把这个传言告诉给 Steve?

(c) 如果有两个人停止相互交流, 那么需要 3 天时间把一则传言从一个人传给其他所有人, 有这样的方法吗?

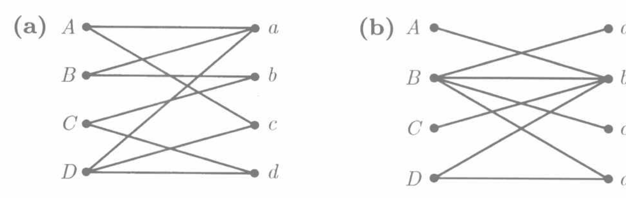
5. (a) 给图 1.4 中的每条边定一个方向, 使得从任意顶点到其他任意顶点存在有向路径.

(b) 重做问题 (a), 要使任意一对顶点间的最长有向路径的长度最小. 并解释为什么不可能得到更小的值.

6. (a) 忽略边的方向时, 图 1.3 中 (有最多顶点的) 最长可能路径的长度是多少?

(b) 图 1.4 中 (有最多顶点的) 最长可能回路的长度是什么?

7. 对于下面各图, 寻找一个匹配或解释为什么不存在匹配:



8. 通过考虑适当的工作子集 (证明它们不可能都被满足), 给出图 1.2 不存在匹配的另一个理由.

9. 通过把匹配定义为图中相邻顶点的配对, 把例 1 中的匹配的概念扩展到任意图. 例如, 图 1.1a 中的一个可能匹配是 $a-b$, $c-d$. 下面各图是否存在? 如果不存在, 解释为什么.

(a) 图 1.4 (b)

10. **(a)** 假设计算机中的词典有一个 “start”, 从它可以检索到 26 个字母中的任意字母, 而且在任意字母处, 你可以达到其前趋和后继字母. 使用图对这一数据结构进行建模.

(b) 另外, 假设你能够从字母 c、k 和 t 返回到 “start”. 现在, 任意两个字母之间的最长有向路径是什么?

11. 在例 2 中构建完整的测试树, 来识别字母表中的 26 个字母.

12. 使用三向比较 (小于、大于或等于), 重复例 2 来识别 26 个字母.

13. 假设花鼠的 8 个当前品种是通过一个进化过程从一个共同的祖先血统演变而来的, 在这个进化过程中的各个阶段, 一个祖先品种分化成两个品种 (祖先品种分化后都不再存活).

(a) 解释如何使用图对这一进化过程进行建模.

(b) 必须发生多少次分化?

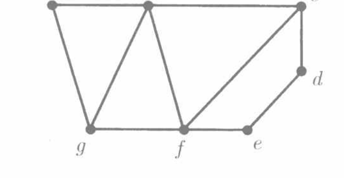
14. 对于例 3, 寻找既不包含边 (a,b) 也不包含边 (b,c) 的边的最小连通集合.

15. **(a)** 在图 1.4 中, 除 (a,b)、(a,e) 和 (c,d)、(d,h) 外, 其余的删除使图断开的 2 边集合是什么? 或者创建这一 2 边集合, 或者给出不存在的理由.

(b) 寻找图 1.4 中删除可以使图断开的所有 2 顶点集合.

16. **(a)** 对于右图, 寻找删除使剩余顶点的图断开的所有 2 顶点集合.

(b) 寻找删除使右图断开的所有 2 边集合.



17. 寻找图 1.2 的最小边覆盖和与其他所有顶点都相邻的顶点的最小集合.

18. 对于图 1.4, 寻找与其他所有顶点都相邻的所有 3 顶点集合.

19. 假设在图 1.4 的网络中加入边 (f,g) 和 (g,j) 并删除 (b,f), 重复例 4, 寻找最小街区监控和街角监控.

20. 假设图 1.4 的网络是由如下大小的南北街和东西街规则阵列构成的, 重复例 4, 寻找边覆盖和最小街角监控.

(a) 南北 3 条街, 东西 3 条街. **(b)** 南北 4 条街, 东西 4 条街.

(c) 南北 5 条街, 东西 5 条街.

21. **(a)** 在棋盘上, 后控制与它在同一行、同一列和同一对角线上的任意方格. 在 8×8 的棋盘上, 至少需要几个后来控制所有方格?

(b) 换成象重复这一问题, 象只控制对角线.

22. 解决图 1.4 的委员会重叠图的会议安排问题. 即, 覆盖所有顶点的独立集合的最小数量是什么?

23. **(a)** 寻找下面各图的最大独立集合:

(i) 图 1.1a; **(ii)** 图 1.2.

(b) 使用 (a) 中的结果, 生成这些图的最小边覆盖.

24. 寻找下面各有向图的顶点基:

(a) 图 1.1b;　(b) 图 1.3;

(c) 图 1.4, 且边的方向由字母表顺序确定 [即, 边 (a, e) 的方向是从 a 到 e].

25. 证明: 有向图中的顶点基是唯一的, 如果在这个图中不存在形成回路的有向边序列.

26. 一个两人游戏从一个空堆开始. 两个玩家轮流把 1 或 2 或 3 个便士放到这个堆中. 谁最后使堆的价值达到 16 便士谁就胜出.

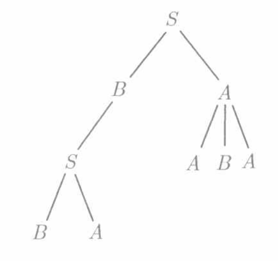
(a) 使用有向图对这一游戏建模.

(b) 证明第二个玩家有必胜策略, 即寻找 4 个 "好" 堆值组成的集合, 包括 16 便士, 使得第二个玩家总是可以移入 "好" 堆中的一个 (当第二个玩家移入一个好堆时, 第一个玩家的下一次移动一定移入一个非好堆, 而从这个位置出发, 第二个玩家便可以移入到一个好堆, 如此反复).

27. 可以使用有向图来表示句子的分析, 其中 (代表整个句子的) 顶点 S 有通向顶点 Su (主语) 和 P (谓语) 的边, 而 Su 和 P 有通向被分解成各成分的部分的边, 以此类推.

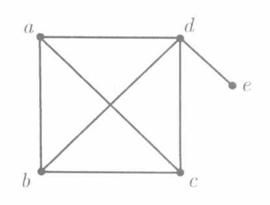
考虑有如下分解规则的抽象语法: $S \to AB$, $S \to BA$, $A \to ABA$, $B \to BAS$, $B \to S$. 例如, $BAABA$ 可以被 "分析" 成如右图. 为下面各句子寻找分析图 (或者解释为什么不存在这样的分析图):

(a) $BABABABA$;　(b) $BBABAABA$.



1.2 同　构

本节研究图的一些基本结构. 我们感兴趣的是把图中的一个顶点与另一个顶点区分开来的性质, 也就是说, 把一个图与另一个图区分开来的性质. 我们带着这样的问题进行讨论: 如何知道画法不同且使用不同顶点名字的两个图实际上是相同的图呢? 例如, 图 1.6 中的两个 5 顶点图是同一个图的两个不同版本吗?



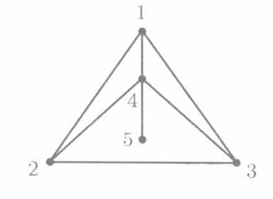


图　1.6

可以用多种不同的方法把一个图画在一张纸上. 因此, 通常有可能用两种方法画一个图, 结果导致观察者认为这些图是 "不同" 的图. 这样的混乱促使我们有下面的定义.

两个图 G 和 G' 称为是**同构** (isomorphic) 的, 如果图 G 中的顶点与 G' 中的顶点之间存在一一对应, 使得一对顶点在 G 中相邻当且仅当这对顶点的对应顶点对在 G' 中相邻.

这样保持相邻顶点间的一一对应称为同构 (isomorphism). 可以如下考虑同构图: 把第一个图重新画在一个透明介质上, 使得这个图可以正好重叠在第二个图的画面上.

如果两个图是同构的, 这两个图必须有相同的顶点数和相同的边数. 图 1.6 中的两个图通过了这一初始检测. 这两个图刚好在一条边的一端分别有一个顶点 e 和 5. 因此, 这两个图的任意同构都必须把 e 与 5 对应. 同样, 从 e 和 5 出发的边的另一端的顶点必须对应, 即 d 与 4 对应. (想象把一张图重叠在另一张图上.) 两个图中剩余的 3 个顶点都彼此相邻 (形成一个三角形), 而且各自与 d 或 4 相邻. 因此 a 与 1、b 与 2、c 与 3 的对应 (或两个 3 顶点子集的任意其他对应) 将保持所需要的相邻关系. 于是, 对应 $a-1$、$b-2$、$c-3$、$d-4$ 和 $e-5$ 是一个同构, 因而这两个图是同构的. 为了形象化地使它们看起来一样, 考虑把右边的图的顶点 4 和 5 向上移动, 然后再向右移 [穿过边 (1, 3)], 使得 1, 2, 3, 4 形成一个有相交对角线的正方形.

回想一下, 一个顶点的度 $\deg(x)$ 是与这个顶点相连的边的数目. 同构保持度不变, 即, 两个对应的顶点必须有相同的度. 于是, 在图 1.6 中, e 必须与 5 对应, 而 d 必须与 4 对应, 因为它们是各自的图中唯一度为 1 和 4 的顶点. 另外, 对于给定的度, 两个同构的图必须有相同的顶点数目. 例如, 如果它们是同构的, 那么图 1.6 中的两个图必须有相等的度为 3 的顶点数. 它们确实如此, 都有 3 个度为 3 的顶点.

一个图 G 的**子图** (subgraph) G' 是由 G 的顶点和边的子集构成的图. 如果两个图是同构的, 那么由对应顶点和对应边构成的子图也一定是同构的. 在图 1.6 中, 删除顶点 e 和 5(以及与其相连的边), 剩下的两个由 4 个彼此相邻的顶点组成的子图是同构的. 一旦我们注意到子图间的这个同构关系, 那么整个图的同构就很容易证明.

可以按下面的方法把子图用于同构检测. 如果图 G 有一个 6 顶点集合, 形成一个长度为 6 的无弦回路 (无弦的意思是除了形成这个回路的这 6 条边之外, 这 6 个顶点之间没有其他边), 那么与 G 同构的任意图也必须有形成这样的无弦 6 回路的 6 顶点集合.

有 n 个顶点且每个顶点都与其他所有顶点相邻的图称为 n 顶点**完全图**(complete graph), 记作 K_n. 2 顶点完全图就是一条边. 从某种意义上讲, 完全子图是所有更大的图的构件. 例如, 图 1.6 中的两个图都是由一个共同的顶点连接到一起的 K_4 和 K_2 组成的. 反过来, 每个 n 顶点图都是 K_n 的子图.

在检测其他图对的同构之前, 我们先提一下确定两个图是否同构的实际意义. 研究有机化合物的学者想要构建之前分析过的化合物的大词典. 当发现一个新化合物时, 他们就想知道这个化合物是否已经在词典中. 大词典中可以有很多这样的化合物, 它们有相同的分子式, 但结构图不一样 (或其他方面不一样). 因此, 他必须

检测这一新化合物, 看看它的图论结构是否与有相同分子式的已知化合物的结构相同 (以及其他方面是否相同), 即, 这种新化合物是否在图论的意义下同构于已知化合物集合中的某一个. 类似的问题也出现在为计算机设计高效集成电路的工作中. 如果同构电路的设计问题已经解决 (或者如果新网络的某个部分与之前已经设计出来的电路同构), 那么就可以节省大量的时间和金钱.

例 1　简单同构

图 1.7 中的两个图同构吗?

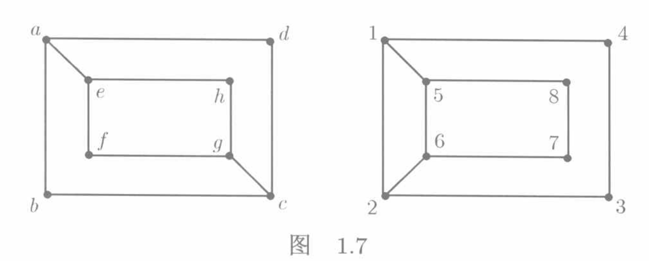


图　1.7

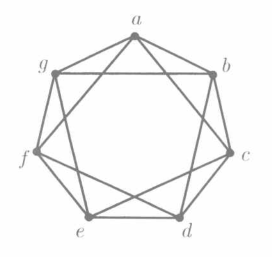
这两个图都有 8 个顶点和 10 条边. 下面检查各顶点的度. 我们看到 d, b, f, h 和 3,4,7,8 的度都为 2, 而其他顶点的度都为 3. 于是这两个图中度为 2 的顶点数相同, 度为 3 的顶点相同. 两个图同构时, 度为 2 的 4 个顶点 (和度为 2 的顶点间的边) 组成的子图必须是同构的. 然而, 顶点 b、d、f 和 h 的任意对之间都不存在边, 而另一个度为 2 的顶点的子图却有两条边 (4,3) 和 (8,7), 所以度为 2 的子图不是同构的, 因此两个完整的图不同构. 读者还可以检测两个图中的度为 3 的子图也不同构.

□

在图 1.7 的左图中, 度为 2 的顶点构成顶点互不相邻的子图. 这样的子图称为**孤立顶点** (isolated vertice) 集合.

例 2　对称图中的同构

图 1.8 中的两个图是同构的吗?



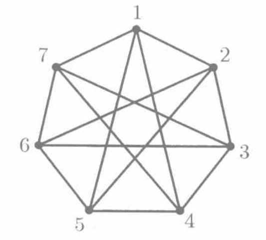


图　1.8

这两个图都有 7 个顶点和 14 条边. 两个图中每个顶点的度都为 4. 另外, 两个图都展示出规则 7 边形的所有对称. 因为在同一个图中顶点间无法区分, 所以我们唯一的选择是尝试着构建一个同构. 从左图的顶点 a 开始. 通过旋转对称, 可以把

a 与右图中的任意顶点相对应 (即, 如果两个图是同构的, 那么将存在一个同构, 使得 a 与右图中的任意给定顶点对应). 我们使用对应 $a-1$.

必须把 a 的邻居 (与 a 相邻的顶点) 集合与 1 的邻居集合对应. 我们来看一下由顶点 a 和 1 的邻居所形成的子图. 参见图 1.9. 这两个子图都是路径：一个是从 f 到 g 到 b 到 c, 而另一条路径是从 7 到 4 到 5 到 2. 这一对应必须使这些路径子图同构. 因此, f 和 c 必须与 7 和 2 对应 (两条路径的端点对应). 根据这个图的左右对称性, f 和 c 与谁对应是没有差异的, 比如说 $f-7, c-2$. 于是, 为了完成 a 和 1 的邻居的同构, 必须把 g 与 4 对应, 把 b 与 5 对应. 现在, 在每个图中只剩下两个没有对应的顶点：d, e 和 $3, 6$ 顶点 g 与 e 相邻, 但不与 d 相邻, 与它对应的顶点 4 与 3 相邻, 但不与 6 相邻. 因此我们必须把 e 与 3 对应, 且 d 与 6 对应.

总之, 根据对称性, 让 a 与 1 对应, f 与 7 对应, 我们得出结论：如果两个图是同构的, 那么一个同构必须是 $a-1, b-5, c-2, d-6, e-3, f-7, g-4$. 检查边, 我们看到这两个图的确由这样的对应而同构 (如果发现这一对应不同构, 那么这两个图一定不同构, 因为除与 a 和 f 的对应相关的对称性外, 我们所做的对应都是强制性的). □

给定图 $G=(V,E), G$ 的**补图** (complement) 是图 $\overline{G}=(V,\overline{E})$, 它与 G 有相同的顶点集合, 但它的边是由 G 中不相连的那些顶点对连接而成. G 与 $\overline{G}$ 中的边的并集构成一个完全图. 两个图 G_1 和 G_2 同构, 当且仅当 $\overline{G}_1$ 和 $\overline{G}_2$ 同构. 利用补图, 例 2 中的同构问题就很容易回答. 图 1.10 给出了图 1.8 中的两个图的补图. 显然, 这两个补图正好是长度为 7 的 (扭曲的) 回路, 因此它们是同构的.

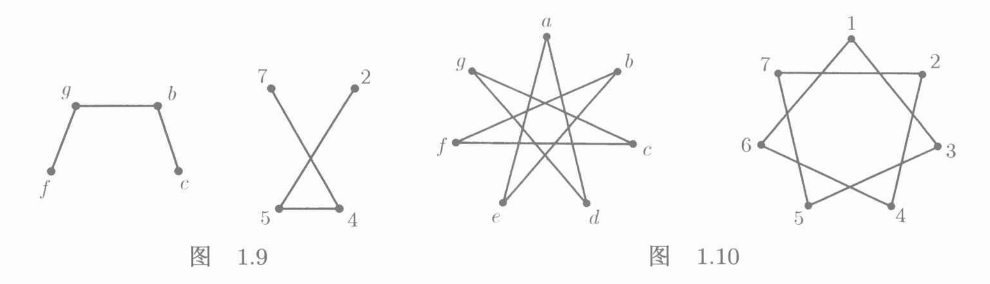


图 1.9　　　　图 1.10

一般来说, 如果一个图中由边连接的顶点对多于没有边连接的顶点对, 那么它的补图将有更少的边, 因此分析起来可能更简单一些.

例 3　有向图的同构

图 1.11 中的两个有向图同构吗?

两个图各有 8 个顶点和 12 条边, 而且每个顶点的的度为 3. 如果把顶点的度分成两部分, **入度** (in-degree)(指向这个顶点的的边数) 和**出度** (out-degree)(离开这个顶点的边数), 则可以看到两个图各有 4 个顶点的入度为 2 而出度为 1, 各有 4 个顶点的入度为 1 而出度为 2. 同前面的例子相同, 我们可以通过从 a 与 1 的对应开

始 (通过对称论证), 然后对应它们的邻居 (边的方向也要对应) 等来构建一个同构.

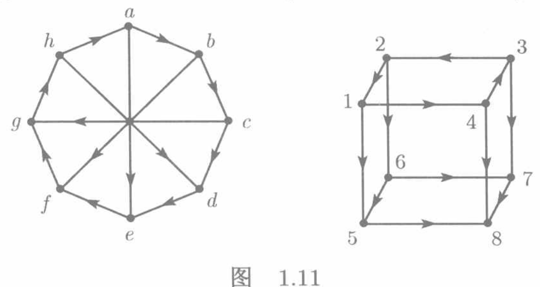


图 1.11

然而, 因为这两个图中的有向路径结构存在根本的差异, 我们将利用这一差异来证明它们不同构. 在左图中, 可以从任意给定的顶点出发到其他任意顶点, 沿着顺时针方向绕着顶点圈来画出一条有向路径: 外面的边构成通过左图中所有顶点的有向回路. 但是在右图中, 顶点子集 $V_1 = \{1,2,3,4\}$ 和 $V_2 = \{5,6,7,8\}$ 之间的所有边的方向都是从 V_1 到 V_2 的, 因此, 没有从 V_2 的顶点到 V_1 的顶点的有向路径 (也不存在经过所有顶点的有向回路). 因此, 这两个图不同构. □

练 习 1.2

1. 列出 4 个顶点构成的所有不同构的无向图.

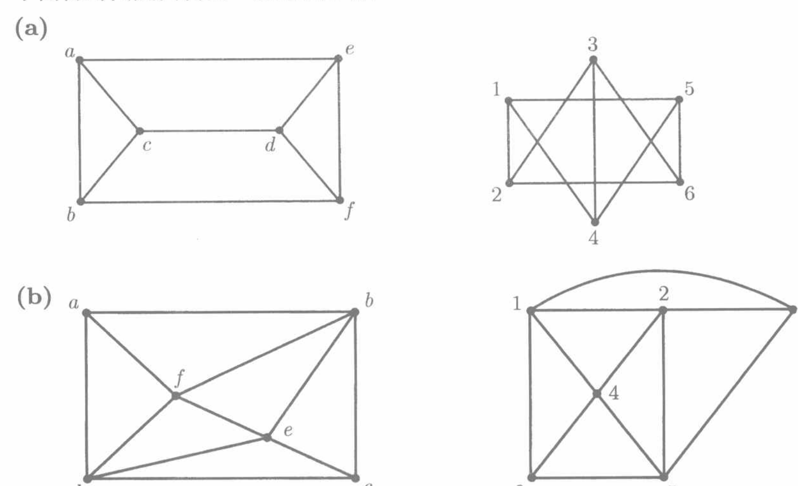
2. 列出 3 个顶点构成的所有不同构的有向图.

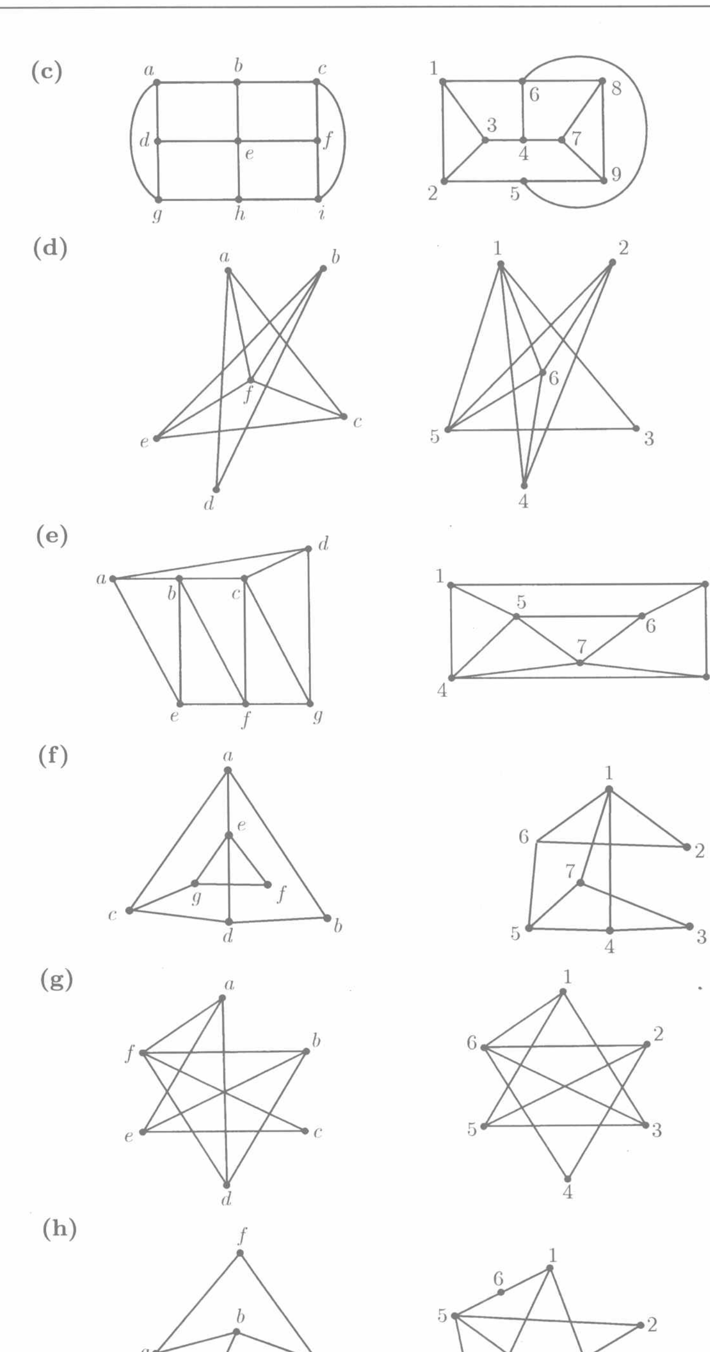
3. 画出两个有下面顶点数和边数的不同构图.

(a) 6 个顶点和 10 条边. **(b)** 9 个顶点和 13 条边.

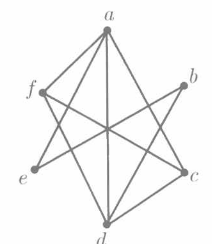
4. 如果忽略方向, 图 1.11 中的两个图同构吗?

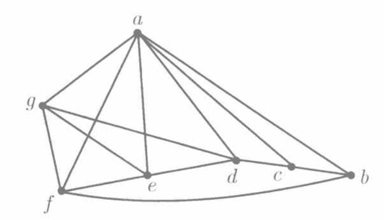
5. 下面哪对图是同构的? 请详细解释.



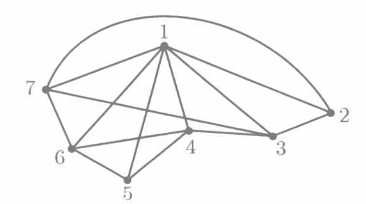

(c)
(d)
(e)
(f)
(g)
(h)

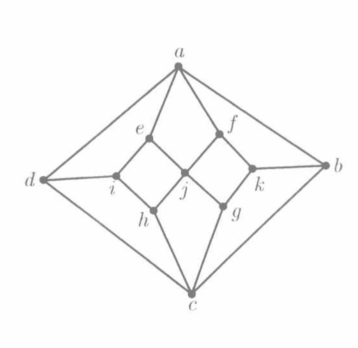
(i)



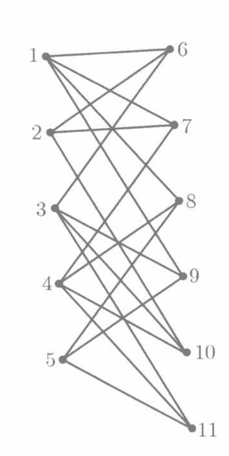


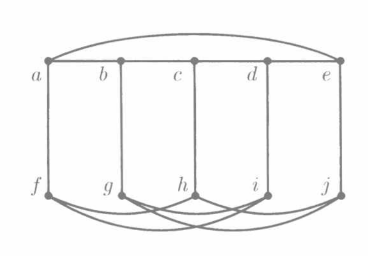
(j)



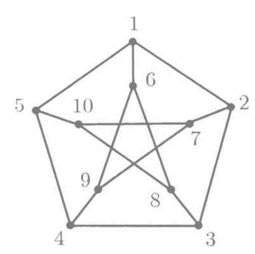


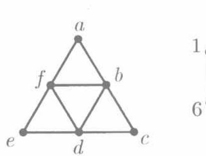
(k)





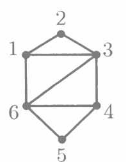
(l)

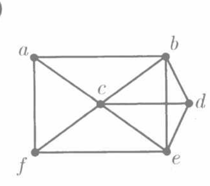




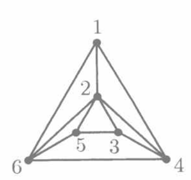
6. 下面哪对图是同构的？请详细解释.

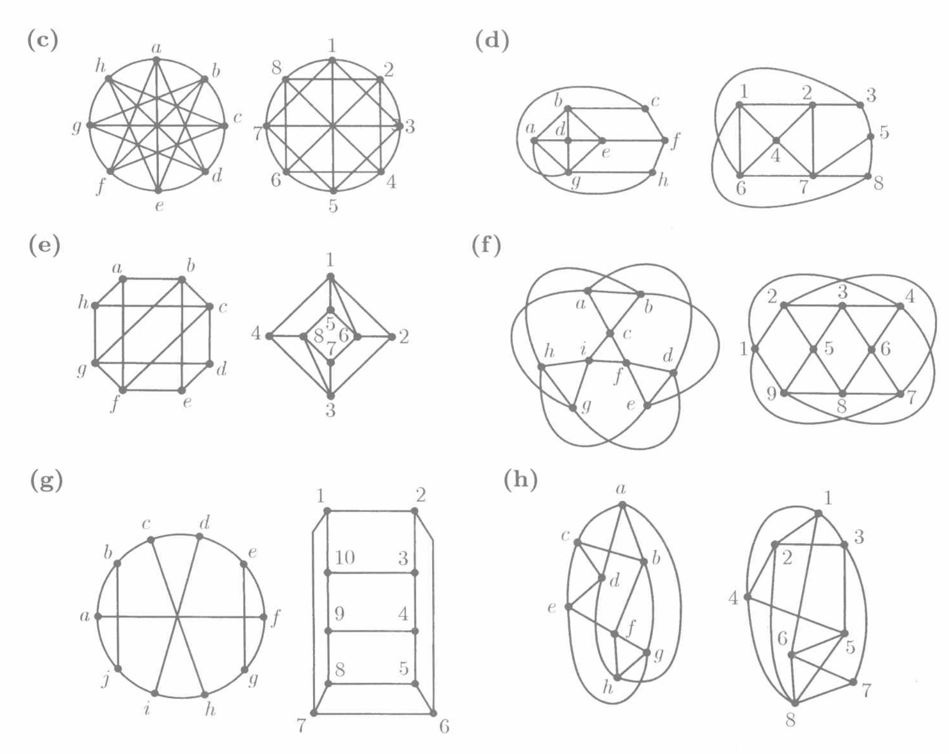
(a)



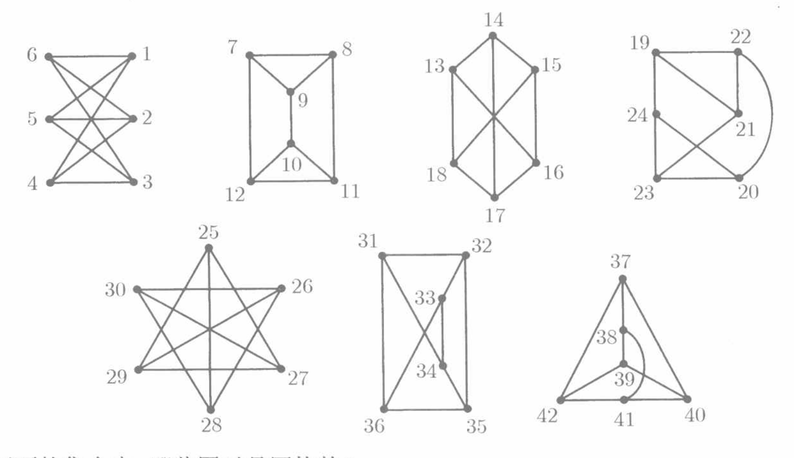


(b)

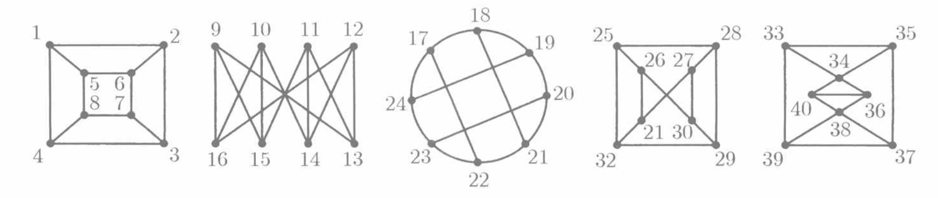




7. 在下面的集合中, 哪些图对是同构的?

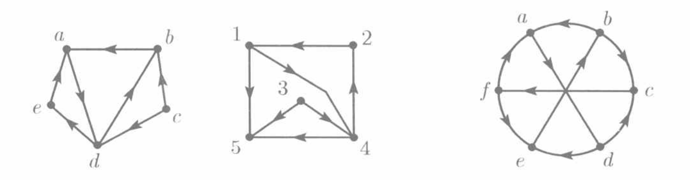


8. 在下面的集合中, 哪些图对是同构的?



9. 假设图 1.8 的每条边的方向都是从较小 (数字或字母) 端的顶点到较大端的顶点. 这样产生的两个有向图同构吗?

10. 下面各有向图对同构吗?



11. 证明: 所有有 5 个顶点且每个顶点的度数为 2 的图都同构.

12. 存在与练习 7 中的某个图不同构且每个顶点度为 3 的 6 顶点图吗?

13. 练习 6(h) 中的两个图的最大完全子图的大小是什么?

14. 如果可能的话, 构建含 6 个顶点且顶点有下面的度的图. 如果不可能, 解释为什么无法构建这样的图.

(a) 有 3 个顶点度为 3, 有 3 个顶点度为 1.

(b) 各顶点的度分别为 1, 2, 2, 3, 4, 5.

(c) 各顶点的度分别为 2, 2, 4, 4, 4, 4.

1.3 边 计 数

关于所有图的一般性结论非常少, 其中一个很有用的一般定理 (或公式) 是边计数.

定理 1 *在任意图中, 所有顶点的度的和等于边数的两倍.*

证明 计算所有顶点度的和时, 计入了与某个顶点连接的某条边的所有实例. 但是, 每一条边都与两个顶点相连, 所以这样的边–顶点相连的总数量等于边数的两倍. 这就证明了此定理. ◇

为了说明定理 1, 考虑图 1.12, 它有 6 个顶点: 有 3 个顶点度为 4, 2 个顶点度为 3, 1 个顶点度为 2. 度的和是 4+4+4+3+3+2=20. 这个和必须等于边数的两倍. 读者可以验证这个图的边数是 10.

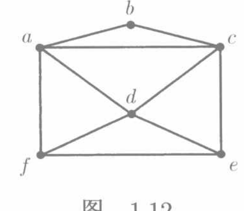


图 1.12

因为度的和是一个偶数, 所以这个和中一定存在偶数个奇数. 因此, 我们得到:

推论 *在任意图中, 度为奇数的顶点个数是偶数.*

下面看一下这个定理和推论的用途.

例 1 定理 1 的用途

假设要构建一个图, 它有 20 条边, 而且所有顶点的度都为 4. 这个图有多少个顶点呢?

设 v 表示顶点数. 顶点的度的和是 $4v$, 根据定理, 这个和一定是边数的两倍: $4v = 2 \times 20 = 40$. 因此, $v = 10$. □

例 2 完全图中的边

n 顶点完全图 K_n 中有多少条边?

回想一下, 在任意可能的顶点对之间, K_n 都有一条边. 在任意的顶点处, 在 K_n 中存在到 $n-1$ 个其余每一个顶点的边, 所以每个顶点的度为 $n-1$. K_n 中所有 n 个顶点的度的和等于 $n(n-1)$. 因为这个和等于边数的两倍, 所以边数是 $n(n-1)/2$. □

例 3 不可能的图

有 7 个人组成一组, 使得其中每个人都正好认识组内其他 3 个人, 这种情况可能吗?

如果用图来对这个问题建模, 顶点代表人, 彼此认识的两个人之间有一条边, 那么我们就创建了一个图, 它有 7 个顶点, 且所有顶点的度为 3. 但是, 根据推论, 度为奇数的顶点数量一定是偶数, 这是不可能的, 所以不存在这样的 7 人集合. □

下面给出一个看似与图无关的谜题. 回想一下, 图 G 是连通的, 如果 G 中的每一对顶点都由 G 中的一条路径连接. 如果 G 不是连通的, 那么可以把它的顶点划分成称为**连通分量** (component) 的连通块. 形式地说, 连通分量 H 是 G 的一个连通子图, 使得 H 中的任意顶点与不在 H 中的 G 的任意顶点之间不存在路径. 包含特殊顶点 x 的 G 的连通分量, 是由 x 以及从 x 通过 G 中的路径可以到达的所有顶点组成的. 注意, 因为 G 的每一个连通分量本身也是一个图, 所以本节的推论既适用于图 G, 也适用于它的每一个连通分量.

例 4 登山者之谜

在顶峰标为 M 的一座山脉的两侧, 两个人分别从相同海拔高度的 A 和 Z 两点出发. 参见图 1.13a. 我们提出下面的谜题: 对这两个人来说, 有没有可能以某种方式沿着图 1.13a 的山脉移动并在 M 点相遇, 并且每一时刻他们都在同一高度上? 我们将证明对于类似于图 1.13a 的任意山脉来说这都是可能的. 所做的一个假设是不存在比 A (或 Z) 低的点, 也不存在比 M 高的点.

我们作一个山脉图 (range graph), 它的顶点是处于同一高度的顶点对 (P_L, P_R), 其中 P_L 在山顶的左边, P_R 在山顶的右边, 使得两点之一是局部峰顶或谷底 (另一个点也可能是局部峰顶或谷底). 我们在图 1.13b 中给出图 1.13a 的山脉图的顶点. 我们做一条边连接顶点 (P_L, P_R) 和 (P'_L, P'_R), 如果这两个人能够在相同的方向上 (两个人都向上行或下行) 分别从点 P_L 到 P'_L, 从 P_R 到 P'_R. 现在, 我们的问题是: 在这个山脉图中是否存在一条从起始顶点 (A, Z) 到山顶顶点 (M, M) 的路径? 对于图 1.13b, 答案显然是肯定的.

我们认为, 在任意山脉图中, 顶点 (A, Z) 和 (M, M) 的度都是 1, 而图中其他所有顶点的度是 2 或 4. (A, Z) 的度是 1, 因为当这两个人从他们各自一侧攀登这座

山时, 他们最初没有选择, 只能往上攀登, 直到有一个人达到一个峰顶. 在图 1.13a 中, 左边遇到的第一个峰顶是 C, 所以从 (A, Z) 有一条到达 (C, X) 的边. 也可以对山顶 (M, M) 进行类似的讨论. 下面考虑顶点 (P_L, P_R), 其中一个点是峰顶, 而另一个点既不是峰顶也不是谷底, 例如 (E, W). 从这一峰顶出发, 可以在两侧向下方行进: 在 W, 我们可以下行到 Z 或者到 U. 两侧的两个人一直下行, 直到有一个人 (或两个人) 到达一个谷底. 对于 (E, W), 两条边分别到达 (D, Y) 和 (D, U). 所以这样的顶点的度是 2. 当一个点 (但不能是两个) 是谷底时, 可以进行类似的讨论. 当顶点 (P_L, P_R) 中的两个点都是峰顶或者都是谷底时, 例如顶点 (D, U), 那么这个顶点的度是 4. 我们把这个证明留给读者作为练习. (由谷底和峰顶组成的顶点的度数是 0. 为什么?)

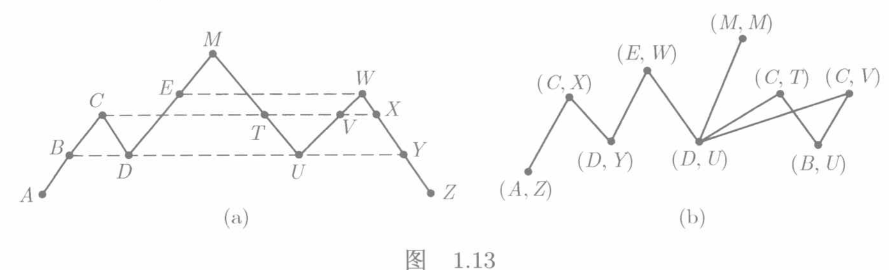


图 1.13

假设在这个山脉图中不存在从 (A, Z) 到 (M, M) 的路径. 我们要使用这样的事实: 起始顶点 (A, Z) 和山顶顶点 (M, M) 是仅有的度为奇数的顶点. 由 (A, Z) 和从 (A, Z) 可以到达的顶点组成的山脉图的连通分量构成正好有一个度为奇数的顶点 (A, Z) 的图. 这与推论矛盾, 所以任意山脉图都一定存在从 (A, Z) 到 (M, M) 的路径. □

图论中很多有趣的性质都依赖于大小为偶数的边集合. 当所有顶点的度都是偶数时, 就引出 2.1 节讨论的欧拉圈.

在 1.1 节的例 1 中, 我们考虑了图 1.14 的匹配问题. 左边的顶点代表人, 右边的顶点代表工作. 从左边顶点到右边顶点的边表示某个人胜任某项工作. 左边两顶点间及右边两顶点间从不存在边. 这样的图称为**二部图**. 从形式上来讲, 图 G 是**二部图**, 如果它的顶点可以分割成两个集合 V_1 和 V_2, 而且每条边连接 V_1 的一个顶点和 V_2 的一个顶点.

二部图可以由这样的事实来刻画: 这种图中的所有回路都有偶长度 (不存在回路的图也是二部图), 其中回路长度或路径长度是回路或路径中边的数量.

定理 2 *图 G 是二部图, 当且仅当 G 中每一条回路有偶长度.*

证明 注意, 只对连通二部图证明此定理就足够了. 我们主张: 如果这一定理对非连通二部图 G 的每一个连通分量为真, 那么它对 G 为真 (在上面的例 4 中给出了连通分量的形式定义). 这一主张得自于这样的事实: G 是二部图当且仅当它的每一个连通分量是二部图, 而且 G 中任意回路有偶长度当且仅当它的每个连通分量

中的每个回路有偶长度.

首先, 我们证明: 如果 G 是二部图, 那么每条回路都有偶长度. 如果 G 是二部图, 我们就可以把图中的所有边画成连接左边顶点与右边顶点的形式. 这时, 对于任意的回路 $x_1 - x_2 - x_3 \cdots x_n - x_1$, 假设第一个顶点 x_1 在左边, 那么回路交替地有一个左边顶点, 一个右边顶点, 再一个左边顶点, 以此类推. 奇数下标的顶点在左边, 而偶数下标的顶点在右边. 参见图 1.15a. 因为 x_n 与 x_1 相邻, 所以 x_n 必须在右边. 所以它的下标是偶数. 即这条回路中的顶点数量是偶数. 回路的边数与顶点数相同, 因此这条回路有偶长度.

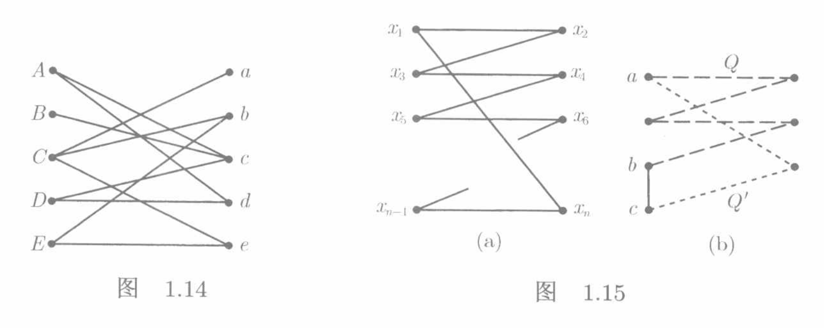


图 1.14 图 1.15

现在, 假设 G 中每条回路 (也可能没有回路) 都有偶长度. 我们说明如何构建一个 G 的二部图布局. 任取一个顶点 a, 把它放在左边. 把所有与 a 相邻的顶点放在右边. 把所有离开 a 两条边的顶点 (即, 距离 a 长度为 2 的路径的端顶点) 放在左边. 一般情况下, 如果在 a 与顶点 x 之间存在奇长度的路径, 则把 x 放在右边. 如果在 a 与 x 之间存在偶长度路径, 则把 x 放在左边.

a 与 x 之间不可能存在分别为奇长度的路径 P 和偶长度的路径 P', 因为从 a 到 x 取路径 P, 然后在 P' 上返回到 a 产生一个奇长度回路. 因为所有回路有偶长度, 所以这是不可能的. (如果 P 和 P' 包含 a 和 x 之外的相同顶点, 那么奇长度回路的存在性证明需要进一步的分析. 详见练习 15.)

同样, 我们证明同在左侧的两个顶点 (比如说 b 和 c) 之间不可能存在边. 否则, 一定存在偶长度路径 Q, Q', 分别把 a 与 b, a 与 c 连接 (因为 b 和 c 都在左侧). 参见图 1.15b, 在这个图中, Q 用破折线表示, Q' 用点线表示. 观察发现, 边 (c, b) 及尾随其后的 Q' 产生一个从 a 到 b 的奇长度路径. 这是不可能的, 因为我们刚刚证明从 a 到 G 中任意其他顶点不可能既存在偶长度路径 (Q) 又存在奇长度路径 $[Q'$ 加上路径 $(a, b)]$. 通过类似的推理, 右边任意两个顶点也不可能是相邻的. 因此, 我们有了一个 G 的二部图布局. ◇

例 5 二部图的检测

图 1.16a 是二部图吗?

任选一个顶点 a, 把它放在左侧. 我们依照定理 2 证明的后半部分的方法, 把

通过偶长度路径与 a 连接的顶点放在左侧, 而把通过奇长度路径与 a 连接的顶点放在右侧. 如果这个图中的所有路径都是偶长度, 那么上面证明中的推理保证这种顶点放置方法将会产生一个二部图布局. 如果我们在左侧 (或右侧) 遇到两个相邻顶点, 那么这个图不可能是二部的. 在这一情况下, 构建成功, 如图 1.16b 所示. □

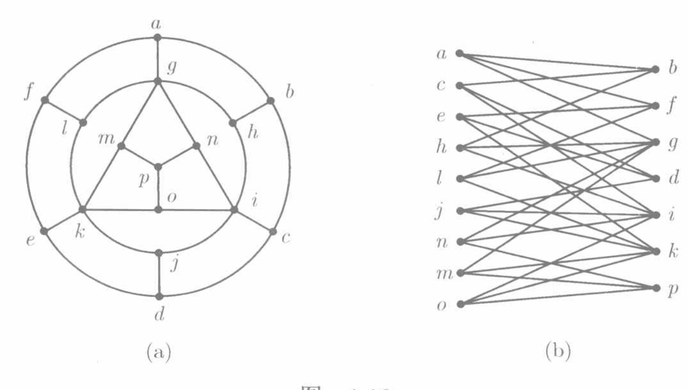


图 1.16

练 习 1.3

1. 符合下面各条件的图有多少个顶点?

(a) 12 条边且所有顶点的度为 2.

(b) 15 条边, 且有 3 个度为 4 的顶点, 其余顶点的度为 3.

(c) 20 条边, 且所有顶点有相同的度.

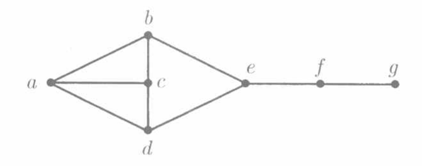
2. 对于下面每一个问题, 描述一个图模型并回答该问题.

(a) 在一个聚会中, 不认识奇数个其他人的人数一定是偶数吗?

(b) 已出生的有奇数个兄弟姐妹的人的数量一定是偶数吗?

(c) 在阿拉斯加, 有奇数个孩子的家庭的数目一定是偶数吗?

(d) 对于下图中的每一个顶点 x, 设 $s(x)$ 表示至少与 x 的一个邻居相邻的顶点 (包括 x) 的数目. $s(x)$ 是奇数的顶点数量一定是偶数吗? 一般情况下这一结论为真吗?



3. 在有 19 条边且所有顶点的度至少为 3 的图中, 顶点的最大可能数目是什么?

4. 如果图 G 有 n 个顶点, 除其中一个顶点外, 其余顶点的度都为奇数. 在 G 的补图 G 中有多少度为奇数的顶点?

5. 证明: 有 m 条边的完全图的顶点数 n 是满足 $n = \frac{1+\sqrt{1+8m}}{2}$ 的整数.

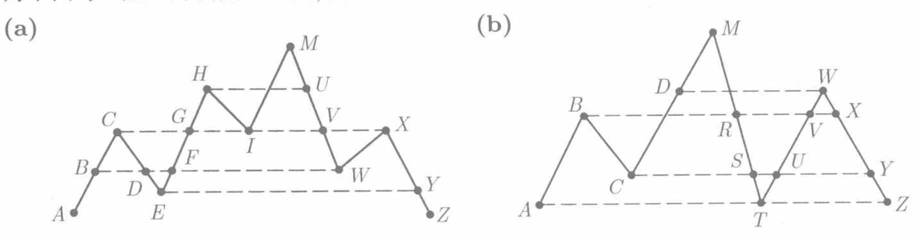
6. 设 G 是有 n 个项点的图, 且它与 $\overline{G}$ 同构. G 有多少条边? (提示: 使用练习 5.)

7. 假设 G 的所有顶点的度都为 p, 其中 p 是奇数. 证明: G 的边数是 p 的倍数.

8. 全美橄榄球联盟 (NFL) 通常由 26 个队组成, 分为两个竞赛联盟, 每个竞赛联盟由 13 个队组成 (每个竞赛联盟又被划分成组, 但这里与此无关). NFL 竞赛指南要求每个队的 14 场比赛应该包括 11 场与所在竞赛联盟中的球队之间的比赛, 以及 3 场与另一个竞赛联盟中的球队之间的比赛. 通过考虑这一调度问题的图模型的右半部分, 证明这一竞赛指南是不可能得到满足的.

9. 证明定理 1 的有向图情形：有向图中顶点的入度之和等于顶点的出度之和, 更进一步, 每个和都等于边数.

10. 为下面每一座山脉构建山脉图, 并使用这个图寻找例 4 中问题的解决方案.



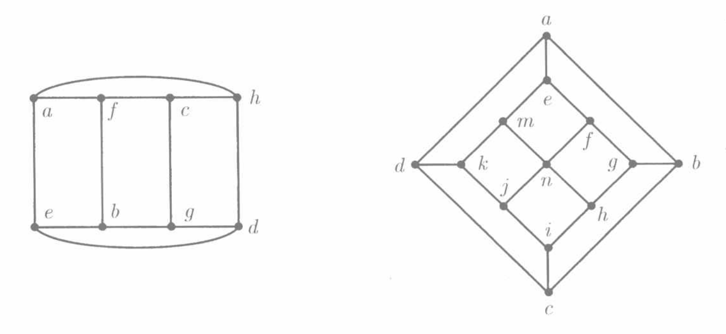
11. 证明：在山脉图中, 如果顶点 (P_L, P_R) 是由两个峰顶或两个谷底组成的, 那么这个顶点的度为 4.

12. 证明：在山脉图中, 如果顶点 (P_L, P_R) 是由一个谷底和一个峰顶组成的, 那么这个顶点的度为 0.

13. 确定下面各图是否二部图. 如果是, 给出如图 1.16b 那样的左边顶点和右边顶点的划分.

(a) 图 1.4. **(b)** 图 1.7(左边的图). **(c)** 图 1.12.

14. 确定下面各图是否二部图. 如果是, 给出如图 1.16b 那样的左边顶点和右边顶点的划分.



15. 假设 x 和 y 是图 G 中仅有的两个度为奇数的顶点, 且 x 和 y 彼此不相邻. 证明：G 是连通的, 当且仅当通过在 G 中添加边 (x, y) 而得到的图是连通的.

16. 在定理 2 的证明的第二部分, 你可能会遇到这样的情况：顶点 a 和 x 之间存在路径 P 和 P', 其中 P 是奇长度, P' 是偶长度, 而且两条路径有共同的顶点. 你必须证明这两条路径上的边的某个子集构成一个奇长度回路. 设 q 是开始于 a 的 P 上 (且同时在 P' 上) 的第一个顶点. 证明：或者从 a 沿着 P 到 q 然后沿 P' 返回到 a 的回路有奇长度, 或者从 q 沿着 P 到 x 然后沿着 P' 返回到 q 的边序列有奇长度. 对于后者, 如果边序列不是回路, 那么它有一个既在 P 上又在 P' 上的顶点 q'. 考虑从 q 沿 P 到 q', 然后再沿 P' 返回到 q 的回路, 或从 q' 沿 P 到 x, 然后沿 P' 返回到 q' 的边序列, 重复上面的推理.

1.4 可平面图

图的最自然例子就是街道地图和电话网络. 来自这样的实体网络的这些图有这样的性质: 可以在纸上描述它们而没有边相交 (不同的边只在顶点相遇). 我们说一个图是**可平面的** (planar), 如果可以在平面上把它画出而没有边相交. 我们使用术语**平面图** (plane graph) 来表示可平面图的平面描述. 图 1.17 中的两个图都是可平面的. 图 1.17b 是一个平面图. 图 1.17a 是可平面图, 因为它可以重新画成图 1.17b 的形式.

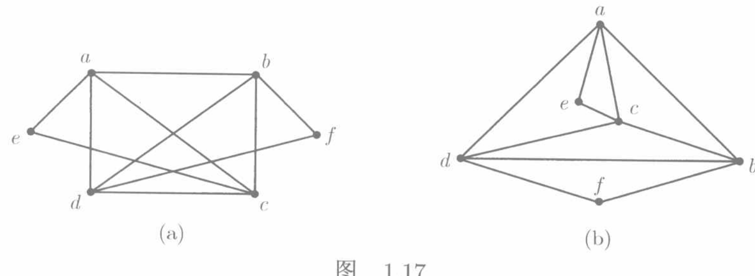


图 1.17

本节的主要目的是确定一个图是否是平面图. 我们采用两种方法. 第一种方法是一个系统方法, 我们设法一条边一条边地画出一个没有相交边的图, 就如同在设法确定两个图是否同构时所做的那样. 第二种方法研究某种理论, 目标就是寻找可平面图的有用性质. 如果一个图不满足这些性质中的一个或多个, 那么我们就知道它不可能是可平面图.

记住, 即便我们已经画出了有相交边的图, 这并不能说明这个图不是可平面图. 也许存在某种方法可以画出这个图, 使得它没有相交边, 就如图 1.17a 所示, 我们可以重新画这个图使其成为图 1.17b.

也许今天对检测一个图是否可平面图的最重要的需求产生于电路设计. 复杂的集成电路是非可平面的, 而且在电路中需要多层 (可平面的) 电路连接. 但是要限制层数, 所以在集成电路设计中一个最重要的问题是把一个大电路分解成最少的已知为可平面图的部分电路.

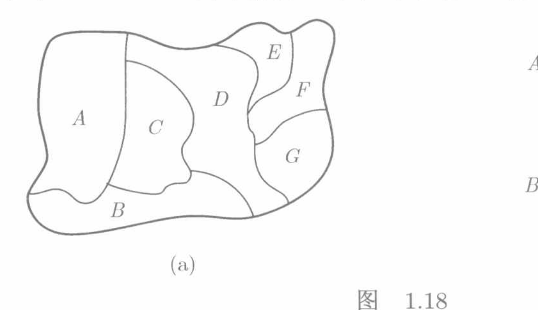
一个更平凡但仍然很重要的平面性检测产生于平面网络中的数据项错误检测. 把城市街道网络这样的一个大可平面图输入到数据终端进行计算机分析时, 通用的错误检测技术首先检测输入的图是否的确是可平面图 (大多数数据项错误会使这个图变成非可平面图).

大约 100 年前, 数学家们在关于图着色问题的研究中首先对可平面图作了广泛的研究.

例 1 地图着色

最著名的数学问题之一是地图着色问题. 这一问题就是: 需要多少颜色来着色

某张地图上的国家, 使得有共同边界的任意一对国家的颜色不同. 国家的地图是以边界为边, 以边界的相交处为顶点的可平面图. 参见图 1.18a. 然而, 与此图有着密切关系, 称为这个图的**对偶图** (dual graph)的可平面图更加有用. 以每个国家为顶点, 而在两个国家之间有共同边界时则在对应顶点间做边, 这样就可以得到它的对偶图. 参见图 1.18b. 通常, 我们把包围地图的无边界区域也当作一个顶点.



(a)　　　　(b)

图　1.18

现在, 对偶图中的问题是: 需要用多少种颜色着色顶点, 使得相邻顶点有不同的颜色. 在图 1.18b 中, 顶点 A, B, C, D 形成 4 顶点完全子图, 所以每个顶点需要一种颜色, 总共需要 4 种颜色. 使用这 4 种颜色, 我们还可以正确地着色剩余顶点.

在过去的一个世纪中, 一个最著名的数学未解问题就是所有可平面图仅用 4 种颜色就可以着色的猜想. 在解决这一猜想的过程中, 数学家发展了大量关于可平面图的理论. 1976 年, Appel 和 Haken 利用大型计算机生成的逐情况、无遗漏分析证明了四色猜想为真 (一共有 1955 种要考虑的图构型情况, 每一种情况又包含若干子情况). 第 2 章将仔细研究图着色问题. □

现在我们尝试着寻找一种系统方法, 以在平面上画一个没有相交边的图. 如两个图的同构那样, 如果构造失败, 那么就会得出该图不是可平面图的结论. 我们称这种方法为**圆弦方法** (circle-chord method). 首先寻找包含图中所有顶点的一个回路 (尽管并不是所有图都存在这样的回路, 但是, 本节所考虑的这类图通常存在这样的回路). 我们把这个回路画成一个大圆. 作为平面图画法, 我们称之为**弦** (chord)的剩余非回路边一定要画在这个圆的圆内或圆外.

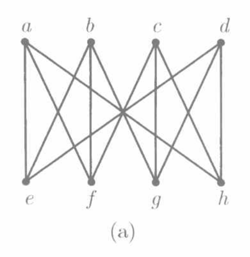
选择第一条弦, 并把它画在圆外. 如果选择正确, 这条弦将迫使某些其他弦被画在圆内 (如果这些弦也放在圆外, 那么它们必定与第一条弦相交). 这些圆内的弦又将迫使某些其他弦被画在圆外, 以此类推. 画好第一条弦后, 我们将被迫选择把后继弦放置在圆内还是圆外. 因此, 如果达到这样一种情况, 无论把一条新弦画在圆内还是画在圆外, 这条新弦都不得不与前面的某条弦相交, 那么, 我们就能说这个图一定是非可平面的. 即基于这个图有可平面描述的假设的构造法导致了一个矛盾. 如果可以在没有相交边的情况下把所有弦都加进来, 那么这个图就是可平面的.

一个关键性的决策是把第一条弦画在圆内还是圆外. 我们认为这没有什么不同, 其原因是下面的圆内–圆外对称性. 考虑两张地图, 第一张地图的中心是北极, 第二张地图的中心是南极. 假设每张地图都有一条路径, 它画在北半球上, 连接赤道上的两个城市. 在第一张地图中 (北半球在赤道以里), 这条路径在赤道圈以内, 而在第二张图中, 这条路径在赤道的外部. 考虑在圆弦方法中以赤道作为回路而形成的圆, 并以两个城市之间的路径作为第一条弦. 这条弦在这个圆内还是圆外恰恰要依你使用哪张地图而定.

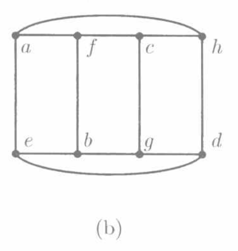
我们注意到, 尽管没有检测两个图是否同构的高效算法, 但存在检测一个图是否可平面图的非常高效的算法. 可平面性检测算法相当复杂, 超出了本书的范围.

例 2 圆弦方法

使用圆弦方法来确定图 1.19a 是否可平面图. 我们寻找一个包含所有 8 个顶点的回路. 一条可能回路是 $a-f-c-h-d-g-b-e-a$. 现在, 尝试把另外 4 条边 (a,h), (b,f), (c,g), (d,e) 加入其中. 由内外对称性, 可以从把边 (a,h) 画到圆外开始. 参见图 1.19b. 这时, 边 (b,f) 和 (c,g) 必须在圆内. 从而 (d,e) 必须在圆外. 所以这个图是可平面图, 如图 1.19b 所示. □



(a)

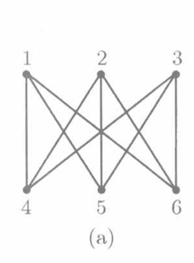


(b)

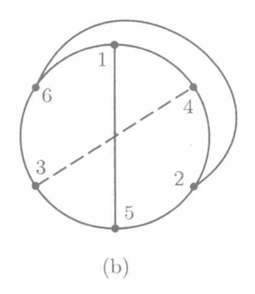
图 1.19

例 3 证明 $K_{3,3}$ 是非可平面图

证明图 1.20a 中 $K_{3,3}$ 是非可平面图. $K_{3,3}$ 的记法表明这个图是一个完全二部图, 它由两个 3 顶点集合组成, 而且一个集合中的每个顶点都与另一个集合中的所有顶点相邻. 运用圆弦方法, 我们在 $K_{3,3}$ 中寻找一个包含所有 6 个顶点的回路, 然后尝试把剩余边 (不在这条回路中的边) 作为内外弦加入其中.



(a)



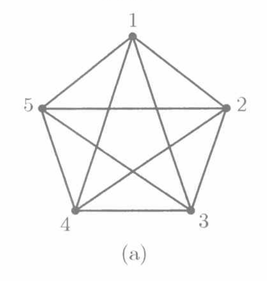
(b)

图 1.20

对于 6 顶点回路, 有若干选择. 假设使用回路 $1-4-2-5-3-6-1$, 并如图 1.20 那样把它画在圆中. 接下来, 必须把边 $(1,5)$, $(2,6)$, $(3,4)$ 加入其中. 首先画弦 $(1, 5)$. 由上面讨论的圆的内外对称性, 可以假设 $(1, 5)$ 被画在回路的内部, 如图 1.20b 所示. 于是 $(2, 6)$ 必须画在回路外面以免与弦 $(1, 5)$ 相交. 最后, 我们必须画边 $(3, 4)$. 如果这条边被画在回路的外面, 则 $(3, 4)$ 一定与弦 $(2, 6)$ 相交; 如果把它画在回路的里面, 那么它一定与弦 $(1, 5)$ 相交. 因此 $K_{3,3}$ 不可能用可平面描述方法画出. 所以 $K_{3,3}$ 是非可平面的. □

利用理论与逐情况分析相结合的办法, 可以证明任何非可平面图包含子图 $K_{3,3}$ 或 K_5 (如图 1.21a 所示的 5 顶点完全图), 或它们略微改动后的图. 使用 (例 3 所使用的) 圆弦方法证明 K_5 是非平面图的工作留作练习. 因此, $K_{3,3}$ 和 K_5 这两个图是一个图能否用平面方式画出的 "原因".

在非可平面性分析中, 我们必须允许对 $K_{3,3}$ 和 K_5 稍加改动. 图 1.21b 给出了一个 $K_{3,3}$ 图, 它在图的某些边的中间加入了一些顶点来对这个图进行*细分*. 其结果图不再是 $K_{3,3}$, 而且也不包含子图 $K_{3,3}$, 但它仍然是非可平面的 (反复把顶点加入到某条边上不能使非可平面图变成可平面图). 称一个子图是一个 *$K_{3,3}$ 构型* ($K_{3,3}$ configuration), 如果可以通过在 $K_{3,3}$ 的某些边上加入顶点而得到这个子图. *K_5 构型* (K_5 configuration) 的定义类似. 下面的可平面图特征定理最初是由波兰数学家 Kuratowski 证明的.



(a)　　(b)

图　1.21

定理 1 (Kuratowski,1930)　*一个图是可平面的, 当且仅当它不包含 K_5 或者 $K_{3,3}$ 构型的子图.*

如果圆弦方法证明一个图是非可平面的, 那么通过定理 1 可知, 这个图有 K_5 或 $K_{3,3}$ 构型子图. 寻找这样的构型有时候也很困难. 然而, 下面的观察很有帮助: *大多数小的非可平面图包含 $K_{3,3}$ 构型.* 练习中除了一个非可平面图外, 其他非可平面图都包含 $K_{3,3}$ 构型. 还要注意, 图 1.20b 所示的有 6 个顶点且有 3 条弦连接相对顶点对的图, 是通常出现于非可平面图中的 $K_{3,3}$ 构型.

例 4　寻找 $K_{3,3}$

使用圆弦方法确定图 1.22a 是否可平面图. 如果它是非可平面图, 那么寻找 $K_{3,3}$ 构型的子图.

首先, 寻找访问所有顶点的回路. 有很多这样的回路. 如图 1.22b 所示, 我们选择回路 $a-b-c-d-e-f-g-h-a$. 假设选择 (a,d) 作为要画的第一条弦. 由内外对称性, 把 (a,d) 画在这个回路的里面和外面没有差异. 我们把它放在回路里面. 于是 (b,e) 必须画在外面以免与 (a,d) 相交. 接下来, 寻找另一个跨过这个圆的弦. 观察发现, 把弦 (c,g) 画在圆内会与 (a,d) 相交, 画在圆外则与 (b,e) 相交. 所以这个图是非可平面图.

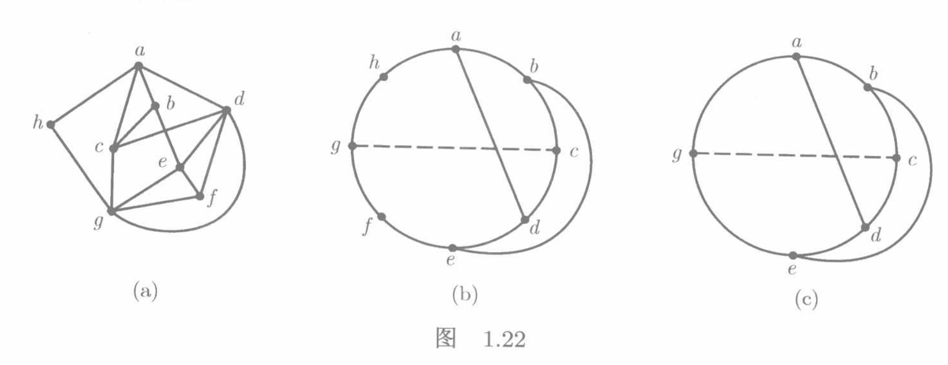


图 1.22

接下来寻找 $K_{3,3}$ 构型. 我们把注意力局限于图 1.22b 中的非可平面子图来简化这个问题, 这个非可平面子图的相交边证明了整个图是非可平面图. 一个 $K_{3,3}$ 构型有 6 个度为 3 的顶点 (对应于 $K_{3,3}$ 的顶点), 外加细化 $K_{3,3}$ 的边的若干度为 2 的顶点. 在子图中寻找 $K_{3,3}$ 构型的方法是消除这个图中的边的细分部分 (删除度为 2 的每一个顶点, 并把这样顶点处的两条边合并成一条边). 图 1.22c 给出了图 1.22b 中的一个消除细分部分后的子图. 图 1.22c 看起来正好是图 1.20b 中的一个 $K_{3,3}$ 的描述. 因此, 图 1.22b 中的这个子图是一个 $K_{3,3}$ 构型. □

在图 1.22a (不使用图 1.22b 中的那个子图) 中寻找 $K_{3,3}$ 构型是很困难的. 在一般的非可平面图中寻找 $K_{3,3}$ 构型的挑战性问题如下所示. 设 z 是原始图中某个度为 3 的顶点, 并假设正好 z 的两条边, 比如说是 (z,r) 和 (z,q), 是一个 $K_{3,3}$ 构型的部分. 这时, z 对应于这一 $K_{3,3}$ 构型的细分顶点, z 的这两条边需要被合并成一条边 (r,q) 来寻找背后的 $K_{3,3}$. 即, z 消失而新边 (r,q) 被创建. 使用圆弦方法所产生的子图能够更加容易地确定 $K_{3,3}$ 构型中度为 2 的顶点, 这样顶点的两条边应该被合并到一起.

可平面图可以有多种不同的平面图描述. 例如, 可以重新画出图 1.23a 中的平面图, 其方法是把由三角形 (d,e,f) 界定的区域做大, 并把顶点 a 放到右侧, 如图 1.23b 所示. 现在, 把这个图的上部和这个三角形的右边翻转到三角形的里面, 得到图 1.23c 的平面图. 三角形 (d,e,f) 变成整个图的外边界. 通过类似的过程, 可以把任意区域的边界转化成整个图的外边界.

尽管可平面图的平面图描述有这种可变性, 但平面图描述的一个重要性质是不变的. 区域的数量总是相同的. 为了简便起见, 假设 G 是连通平面图. (回想一下,

连通意味着每一对顶点间都存在路径.) 如果 v 和 e 分别表示 G 中的顶点数和边数, 那么 G 的平面图描述将总是有 r 个区域, r 由公式 $r=e-v+2$ 给出. 记住, 图外面的无界区域计为一个区域. 这个关于 r 的重要公式是欧拉在大约 250 年前发现的.

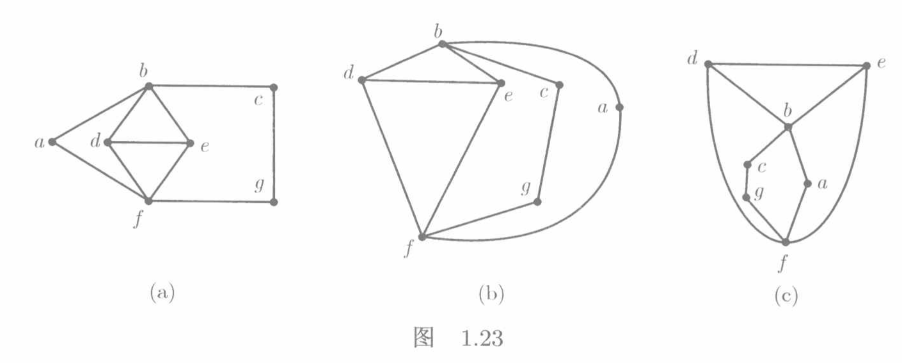


图 1.23

定理 2 (欧拉公式, 1752) 如果 G 是连通平面图, 那么 G 的任意平面图都有 $r=e-v+2$ 个区域.

证明 我们一条边一条边地画出 G 的一个平面图描述. 依次选择边, 使得在每一个阶段都得到一个连通子图. 设 G_n 表示把 n 条边加入之后所得到的平面图, 并设 v_n、e_n、r_n 分别表示 G_n 中的顶点数、边数和区域数. 开始时我们有 G_1, 它是由一条边、这条边的两个端顶点以及一个 (无界) 区域组成的. 于是 $e_1=1, v_1=2, r_1=1$, 所以欧拉公式对 G_1 是正确的, 因为有 $r_1=e_1-v_1+2: 1=1-2+2$. 通过在 G_1 的一个顶点处加入一条边, 我们从 G_1 得到 G_2. 一般说来, G_n 是通过在 G_{n-1} 的一个顶点处加入第 n 条边而得到的. 这条新边可能是连接已在 G_{n-1} 中的两个顶点. 否则, 这第 n 条边的另一个端顶点也要加入到 G_n 中, 成为其中的一个新顶点.

现在, 使用归纳法 (参见附录 A.2) 完成证明. 已经证明定理对 G_1 为真. 接下来, 对于任意的 $n>1$, 假设定理对 G_{n-1} 为真并证明它对 G_n 也为真. 设 (x,y) 是加入到 G_{n-1} 得到 G_n 的第 n 条边. 要考虑两种情况.

第一种情况是 x 和 y 都在 G_{n-1} 中. 这时, 它们在 G_{n-1} 的同一个区域 K 的边界上, 这一区域也可能是无界区域 [如果 x 和 y 不在同一区域, 那么不可能以平面描述的形式画出所需的边 (x,y)]. 参见图 1.24a. 边 (x,y) 把区域 K 分成两个区域. 因此, $r_n=r_{n-1}+1, e_n=e_{n-1}+1, v_n=v_{n-1}$. 所以欧拉公式的两边各增加 1. 因此, 如果这个公式对 G_{n-1} 为真, 那么它对 G_n 也为真.

第二种情况是 x 和 y 中有一个不在 G_{n-1} 中, 比如说 x 不在 G_{n-1} 中. 参见图 1.24b. 这时, 加入边 (x,y) 意味着把 x 也加入其中, 但是不形成新区域 (即, 现存区域没有被分割). 因此, $r_n=r_{n-1}$, $e_n=e_{n-1}+1$, $v_n=v_{n-1}+1$, 欧拉公式的两边的值没有变化. 所以欧拉公式对 G_{n-1} 为真意味着它对 G_n 也为真.

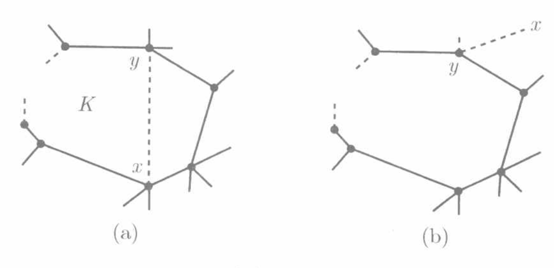


图 1.24

所以, r 的每一次增加都因 e 或 v 的增加而在欧拉公式中被平衡掉. 根据归纳法, 公式对 G_n 为真, 因此对整个图 G 为真. ◇

例 5 使用欧拉公式

有 10 个顶点且每个顶点的度为 3 的平面图有多少个区域?

根据 1.3 节的定理 1, 度的总和为 10×3, 它等于 $2e$, 所以 $e = 15$. 根据欧拉公式, 区域数 r 是

$$r = e - v + 2 = 15 - 10 + 2 = 7$$

□

定理 2 有下面的推论, 我们经常使用这个推论来快速证明一个图是非可平面图.

推论 *如果 G 是连通可平面图且 $e > 1$, 则 $e \leqslant 3v - 6$.*

证明 类似于顶点的度的定义, 我们定义一个区域的度为这个区域所附带的边数, 即, 在它的边界上的边数. 如果一条边沿边界出现两次, 如图 1.25a 中区域 K 中的边 (x, y), 那么这条边在 K 的度中计数两次; 例如, 在图 1.25 中, 区域 K 的度为 10, 而区域 L 的度为 3. 观察发现, 平面图的每个区域的度必须大于等于 3, 因为度为 2 的区域由连接相同顶点对的两条边界定, 而对于度为 1 的区域则由一条循环边界定 (参见图 1.25b), 但是在图中不允许有平行边和循环.

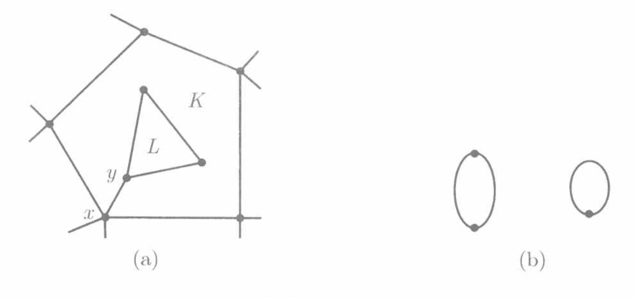


图 1.25

因为平面图的每个区域的度都大于等于 3, 所有区域的度之和至少是 $3r$. 但是所有区域的度的和必须等于 $2e$, 因为这个和把每一条边计数两次, 即, 每条边的两侧都是某个边界的一部分 (这一讨论与用于证明顶点的度的和等于 $2e$ 相同). 因此,

$2e$=(区域的度的和) $\geqslant 3r$, 或者 $2/3e \geqslant r$. 把这一不等式与欧拉公式 (定理 2) 结合起来, 我们有

$$\frac{2}{3}e \geqslant r = e - v + 2 \quad \text{或者} \quad 0 \geqslant \frac{1}{3}e - v + 2$$

求解 e, 我们得到 $e \leqslant 3v - 6$. ◇

例 6 证明 K_5 是非可平面图

使用推论, 证明 5 顶点完全图 K_5 是非可平面图.

图 K_5 有 $v = 5, e = 10$ (寻找完全图的边数的方法, 见 1.3 节的例 2). 于是 $3v - 6 = 3 \times 5 - 6 = 9$. 但是推论说在连通平面图中 $e \leqslant 3v - 6$ 一定为真, 所以 K_5 不是可平面图. □

这个推论*不应错误地理解为*: 如果 $e \leqslant 3v - 6$, 则连通图是可平面图. 很多非可平面图也满足这个不等式. 例如, 对于 $K_{3,3}$, 就有 $v = 6, e = 9$, 满足这个不等式.

两个定理和推论奠定了可平面图数学理论的基础. 在这样的过程中, 我们获得了证明一个图是非可平面图的实用工具. 扩展这一理论的一种方法是使推论中的不等式变得 "更强", 即, 得到 e 的一个更小上界. 回想一下, 证明这个推论的最关键步骤是观察每个区域的度至少为 3. 这导致不等式 $2e \geqslant 3r$. 假设某个 (至少有两条边的) 连通图 G 已知是二部图. 根据 1.3 节中的定理 2, 这个二部图中的所有回路有偶长度. 这时, 这个图中的任意区域的度都不能是 3(因为这将意味着存在长度为 3 的边界回路). 于是, 二部可平面图中的每个区域的度一定 $\geqslant 4$. 对所有区域的度求和, 我们得到不等式 $2e =$ (区域的度数和)$\geqslant 4r$. 使用不等式 $2e \geqslant 4r$ 重新推导推论, 我们有 $2/4e \geqslant r = e - v + 2$, 因此, 有

$$e \leqslant 2v - 4$$

每个二部连通可平面图必定满足这个不等式. 考虑我们 "喜欢的" 二部图 $K_{3,3}$. 它有 $v = 6, e = 9$, 如上所述, 它满足推论的不等式 $e \leqslant 3v - 6$. 但是因为它是二部图, $K_{3,3}$ 如果是可平面图的话, 它还应该满足新的不等式 $e \leqslant 2v - 4$. 这不成立: $9 > 2 \times 6 - 4$.

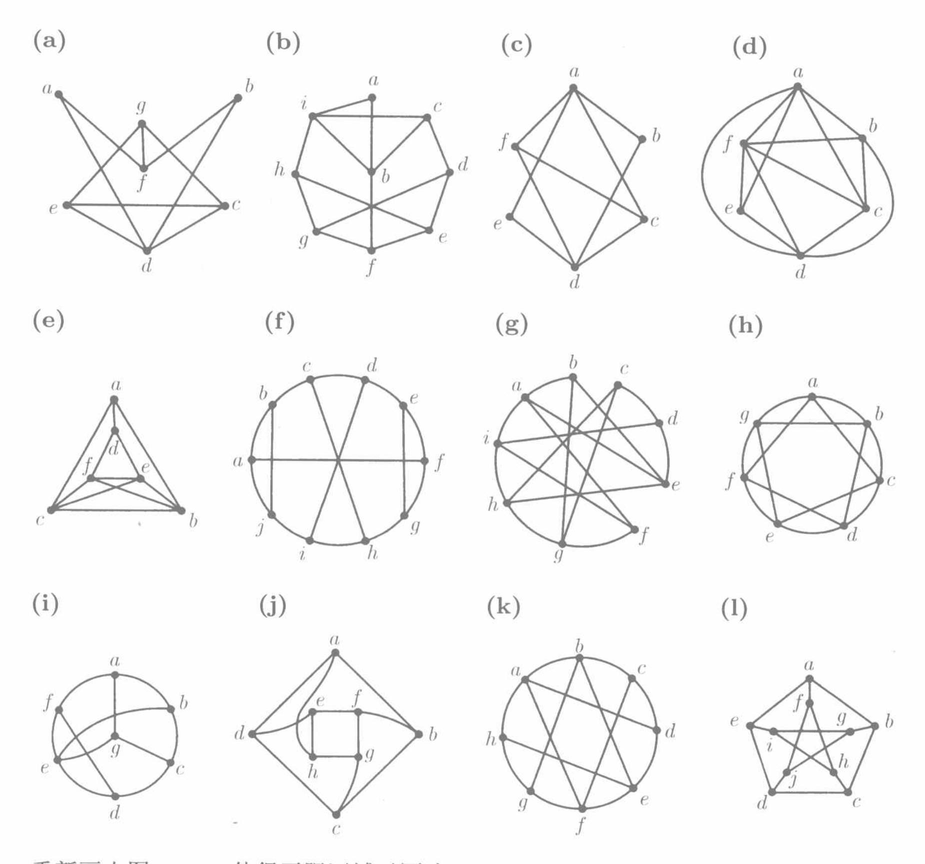
练 习 1.4

练习概况 前 6 道练习确定各种图是否可平面图, 并用不同方法画出可平面图. 练习 10 涉及对偶. 练习 15~26 基于欧拉定理和推论 $e \leqslant 3v - 6$. 其余练习介绍新概念.

1. 画出下面各图中的可平面图的对偶图 (包括对应于图外部的无界区域的顶点):
(a) 图 1.17b; **(b)** 图 1.18b.

2. 利用例 2 的方法, 证明 K_5 是非可平面图.

3. 下面各图中哪些是可平面图? 在非可平面图中寻找 $K_{3,3}$ 或 K_5 构型 (几乎都是 $K_{3,3}$ 构型).



4. 重新画出图 1.19b, 使得无限区域以回路 $a-f-c-h-a$ 为边界.

5. **(a)** 对于什么样的 n 值, K_n 是可平面图?

(b) 对于什么样的 r 和 s 值, 完全二部图 $K_{r,s}$ 是可平面图? ($K_{r,s}$ 是 r 个顶点在左侧, s 个顶点在右侧, 且在左侧与右侧的所有顶点对间有边的二部图.)

6. 完全三部图 $K_{r,s,t}$ 是完全二部图的扩展 [参见第 5 题练习的部分 (b)]. $K_{r,s,t}$ 有 3 个顶点子集, 第一个子集中有 r 个顶点, 第二个子集中有 s 个顶点, 第三个子集中有 t 个顶点. 每个顶点子集中的每一个顶点都与其他两个子集中的每一个顶点相邻, 即, 每个顶点与不在同一个子集中的所有顶点相邻, 确定 $K_{r,s,t}$ 是可平面图的所有三元组 r,s,t.

7. 对于下面每一种情况, 假设图是可平面的, 给出 r,e,v 的值. 然后, 如果存在的话, 画出有该性质的连通可平面图, 否则解释为什么不存在这样的可平面图.

(a) 8 个顶点和 13 条边. **(b)** 6 个顶点和 7 个区域.

(c) 14 个顶点和 9 个区域. **(d)** 6 个顶点和 14 条边.

(e) 5 个区域和 10 条边. **(f)** 6 个顶点, 度都为 4.

(g) 7 个顶点, 度都为 3. **(h)** 5 个区域, 每个区域有 4 条边界边.

(i) 12 个顶点, 每一个区域有 4 条边界边.

(j) 17 个区域, 每个顶点的度都为 5.

8. 如果有 n 个度都为 4 的顶点的连通可平面图有 10 个区域, 确定 n.

9. 图 G 的线图 (line graph) $L(G)$ 是这样的图：它以 G 的每条边为顶点，而且对于 G 中每两条有共同端点的边，$L(G)$ 有一条连接对应的顶点对的边.

(a) 证明：$L(K_5)$ 是非可平面图.

(b) 寻找其线图是非可平面图的可平面图.

10. 可以把对偶 $D(G)$ 的构造法运用于任意平面图 G：对 G 的每一个区域，画出 $D(G)$ 的一个顶点，并在 G 的每条边 e 的垂直方向上做 $D(G)$ 的边 e^*；e^* 连接代表 e 两侧区域的 $D(G)$ 的顶点.

(a) 图的对偶不必是图. 它可能在相同顶点对之间有两条边，或有 (从一个顶点到其自身的) 自循环边. 寻找对偶不是图的两个可平面图，它们分别包含这两种不允许出现的情况.

(b) 证明：图 1.23a 和 1.23c 的两个不同平面描述的对偶是同构的.

(c) 证明：对偶图 $D(G)$ 中的顶点的度等于可平面图 G 中对应区域的边界的边数.

(d) 寻找一个与其对偶同构的可平面图.

(e) 对于图 G 的任意平面表述，证明 G 的顶点对应于 $D(G)$ 的区域.

11. (a) 证明：如果可平面图中的一条回路正好围绕两个区域，其中每个区域有偶数个边界边，那么这条回路有偶长度.

(b) 证明：如果可平面图中的一条回路围绕一组区域，其中每个区域有偶数个边界边，那么这条回路有偶长度.

12. 图 G 的相交数 (crossing number) $c(G)$ 是 G 的描述中相交边对的最小数. 例如，如果 G 是可平面图，那么 $c(G) = 0$. 对于下面各图，确定 $c(G)$：

(a) $K_{3,3}$； **(b)** K_5； **(c)** 图 1.22a.

13. 图 G 是临界非可平面的 (critical nonplanar)，如果 G 是非可平面图，但是删除任意顶点得到的子图是可平面图.

(a) 下面各图中哪些是临界非可平面图？

(i) $K_{3,3}$； **(ii)** K_5； **(iii)** 图 1.22a.

(b) 证明：临界非可平面图一定是连通的，而且不包含这样的顶点，它的删除断开这个图.

14. 图 G 称为最大可平面的 (maximal planar)，如果 G 是可平面图，但是在两个不相邻顶点间加入另一条边将导致一个非可平面图.

(a) 证明：连通的最大可平面图的每个区域是三角形.

(b) 如果连通最大可平面图有 n 个顶点，那么它有多少个区域和多少条边？

15. 假设 G 是可平面图但不一定如欧拉公式 (定理 2) 所要求的那样是连通的. 回想一下，连通分量 H 是图 G 的连通子图且有这样的性质：H 中的任意顶点与不在 H 中的 G 中任意顶点之间不存在路径.

(a) 对有 c 个连通分量的可平面图，寻找欧拉公式的相应修改版本.

(b) 证明：对于非连通可平面图，推论成立.

16. 证明：如果 G 有 11 个顶点，那么或者 G，或者它的补 $\overline{G}$ 是非可平面的. (提示：确定 11 顶点完全图的总边数 N_{11}；如果结果为假，即 G 和它的补都是可平面图，那么这两个图各包含 N_{11} 中的多少条边？)

17. 仿照推论的讨论，证明在每个顶点的度都大于 3 的可平面图中，$e \leqslant 3r - 6$.

18. (a) 证明：每个连通可平面图都有一个度至多为 5 的顶点. [提示：假设每个顶点的度至少

为 6, 如下所示, 将得到一个矛盾. 得到一个涉及 v 的关于顶点度的总和的下界 (类似于在推论的证明中得到的区域度的和的下界). 因为这个度的和等于 $2e$, 我们可以使用推论中 e 的边界得到涉及 v 的关于这个度的和的一个上界. 把这两个界结合起来得到矛盾.]

(b) 证明: 问题 (a) 可以直接扩展到任意 (非连通) 可平面图.

19. 证明: 每个顶点数小于 12 的连通可平面图都有一个度至多为 4 的顶点. [提示: 假设每个顶点的度至少为 5, 得到一个 e 的下界 (再结合推论中的 e 的上界), 由此推导出 $v \geqslant 12$.]

20. 如果 G 是回路长度至少为 k 的连通可平面图, 证明不等式 $e \leqslant 3v-6$ 可以加强成 $e \leqslant \frac{k}{k-2}(v-2)$. (提示: 区域的度至少是 k.)

21. (a) 证明: 对于练习 3(1) 的图, 每条回路的度至少为 5.

(b) 使用问题 (a) 和练习 20 的结果, 证明这个图是非可平面图.

22. (a) 给出一个区域只有方形 (区域以 4 条边为边界) 和六边形组成的, 且顶点的度至少为 3 的图.

(b) 用 v 以及方形数 s 写出在这样的图中顶点的度的和 ($=2e$) 的表达式. 然后利用练习 17 得到 $2e$ 的上界. 推断在问题 (a) 定义的这类图中至少有 6 个方形.

(c) 如果每个顶点的度都为 3, 证明问题 (a) 定义的这类图正好有 6 个方形.

23. 如果 G 是连通可平面图, 其中 $e=3v-6$, 证明每个区域都是三角形 (有 3 条边界边).

24. **柏拉图图** (platonic graph) 是一个可平面图, 其中所有顶点有相同的度 d_1, 而所有区域有相同的边界边数 d_2, $d_1 \geqslant 3$ 且 $d_2 \geqslant 3$. 柏拉图图是柏拉图立方体 (如八面体) 的 "骨架".

(a) 设 G 是柏拉图图, 且分别有顶点度 d_1 和区域度 d_2, 证明: $e = 1/2d_1v$ 且 $r = (d_1/d_2)v$.

(b) 使用问题 (a) 和欧拉公式, 证明: $v(2d_1+2d_2-d_1d_2)=4d_2$.

(c) 因为 v 和 $4d_2$ 是正整数, 所以我们从问题 (b) 得出结论: $2d_1+2d_2-d_1d_2>0$. 利用这个不等式证明: $(d_1-2)(d_2-2)<4$.

(d) 从问题 (c) 出发, 寻找 5 对可能的 d_1, d_2 的正 (整数) 值.

25. 假设画 l 条穿过一个圆的直线, 且这些直线形成 p 个交点 (使得正好两条直线相交于一点). 由这些直线在圆内形成的区域数 r 是多少? 假设这些直线在圆的边上形成 $2l$ 个不同的交点.

26. 考虑 4 个圆 A, B, C, D 所形成的重叠集合. 我们希望定位这些圆, 使得这些圆的每个可能的子集形成一个区域, 即, 每一个都正好包含在一个圆中的 4 个区域, 由两个圆的交形成的 6 个区域 (AB, AC, AD, BC, BD, CD), 由 3 个圆的交形成的 4 个区域, 由所有 4 个圆的交形成的 1 个区域. 证明: 不可能有这样的 15 个有界区域的集合.

27. 证明: 可以把下面各图画在一个圆环图 (doughnut) 的表面, 使得边不相交.

(a) $K_{3,3}$. **(b)** K_5. **(c)** K_6.

1.5 小结及参考文献

本章介绍了图、图的应用以及一些基本结构. 我们采用一种面向用户的方法来处理图论. 对作为纯数学的更加形式化的图论感兴趣的读者, 应该看一看 Bondy and Murty[1]、Harary[3] 和 Wilson[4]. 1.1 节介绍了一组说明性的图模型. 1.2 节通

过确定使两个图不同的原因探讨了图的基本结构. 1.3 节给出了若干有用的边计数结果. 最后一节介绍了几类重要的可平面图. 我们概述了确定一个图是否可平面图的实用方法和理论方法.

图论的历史开始于 1736 年欧拉 (L. Euler) 关于欧拉圈 (将在 2.1 节讨论) 的工作. 最初用多面体描述的可平面图的欧拉公式是在 1752 年得到证明的. 在关于拓扑学和几何游戏的一些论文中出现了一些图的理论, 但是直到 1850 年前后, 人们才开始对图进行正式的研究. 研究之一是凯莱 (A. Cayley) 于 1850 年发表的论文, 讨论了树的计数 (将在第 3 章讨论). 另一个研究是基尔霍夫 (G. Kirchhoff) 于 1847 年发表的论文, 给出一种回路代数并在电路研究中引入了图. 该论文还包含了基尔霍夫著名的电流、电压定律 (基尔霍夫写下这篇历史性的论文时只有 21 岁). 图这个术语是由西尔维斯特 (J. Sylvester) 于 1877 年首先开始使用的. 图论的第一部著作是由 D. Konig 编著的, 直到 1936 年才得以出版. 极其优秀的有关图论历史的原始资料是图论 1736–1936, 见文献 [2].

关于图论内容的其他文献, 参见本书最后的参考文献.

1. J. A. Bondy and U. S. R. Murty, *Graph Theory with Applications*, American Elsevier, New York, 1976.

2. N. Biggs, E. Lloyd, and R. Wilson, *Graph Theory 1736–1936*, Cambridge University Press, Cambridge, 1999.

3. F. Harary, *Graph Theory*, Perseus Press, New York, 1995.

4. R. Wilson, *Introduction to Graph Theory*, 4th ed., John Wiley and Sons, New York, 1997.

补 充 练 习

练习概况 图论因其有趣的问题而著称. 若干练习介绍新的图概念, 例如, 强连通性 (练习 14) 和割集 (练习 25). 练习 30 是拉姆齐 (Ramsey) 理论中的一个非常著名的问题 (参见附录 A.4). 对于图论中更多有趣的问题, 参见本书末尾参考文献所列出的图论书籍.

1. 假设有 7 个委员会, 且每一对委员会有一名共同的成员, 每名成员同处于两个委员会. 7 个委员会共有多少人?

2. 证明: n 顶点完全图 K_n 的补图是 n 个孤立顶点的集合.

3. 一个图是*正则的* (regular), 如果所有顶点有相同的度. 设一个有 n 个顶点的图是顶点度为 3, 且有 18 条边的正则图, 确定 n.

4. 设一个图有 50 条边, 它的最少顶点数是多少?

5. 证明: 在至少有两个顶点的任意图中, 至少有两个顶点有相同的度. (*提示*: 注意度为 0 的顶点.)

6. 证明: 所有顶点的度都至少为 2 的无向图一定包含回路 (在回路中, 边不能重复).

7. 设图 G 中每个顶点的度为 2, G 的每个顶点都在一条回路上吗? 证明或给出反例.

8. 设图 G 中每个顶点的度都至少为 d, 证明 G 一定包含长度至少为 $d+1$ 的回路.

9. 设有向图 G 中的每个顶点的出度都是正数 (至少有一条向外的有向边), 这时:

(a) G 一定包含有向回路吗?

(b) G 的每个顶点一定在一条有向回路上吗?

10. 如果 G 是一个非完全的连通图, 证明某个顶点 x 有两个互不相邻的邻居 y, z [即, 存在边 (x, y) 和 (x, z), 但是不存在边 (y, z)].

11. 证明: 非连通图的补图一定是连通的.

12. 证明: 顶点 x 的删除断开连通无向图 G, 当且仅当 G 中有两个顶点 a 和 b 使得 G 中所有从 a 到 b 的路径都经过 x.

13. 设 G 是一个连通图, 使得对于除 G 中两个顶点外的所有顶点 x, $G-x$ 是非连通的. 证明 G 是一条路径.

14. 有向图 G 称为是*强连通的* (strongly connected), 如果对于 G 中的任意两个顶点 x 和 y, 存在从 x 到 y 的路径. 给下面的图的边标上方向使之成为强连通图. 如果不可能, 解释为什么. (这个问题的一个应用是使一个城市的街道成为单行道.)

(a) 图 1.4. (b) 图 1.2.

15. 证明: G 是强连通的 (参见练习 14), 当且仅当 G 的顶点不能被分割成为两个集合 V_1, V_2, 使得不存在从 V_1 的顶点到 V_2 的顶点的边.

16. 桥(bridge) 是连通图中的一条边, 它的删除将使这个图不再连通. 证明: 如果图 G 包含桥, 那么它没有包含 G 的所有顶点的回路.

17. 证明: 如果连通图 G 的每条边都在一个回路上, 那么 G 不可能有桥 (练习 16 中定义了桥).

18. 证明: 可以为连通无向图 G 的边定向而创建一个强连通图 (参见练习 14), 当且仅当 G 中没有桥 (参见练习 16).

19. 如果有可能, 画出有下面各顶点类型的平面图:

(a) 6 个度为 3 的顶点; (b) i 个度为 i 的顶点, $i = 1, 2, 3, 4, 5$;

(c) 两个度为 3 的顶点和 4 个度为 5 的顶点.

20. Megabucks 先生邀请 3 对夫妇到他的公寓赴晚宴. 客人到达后, Megabucks 先生和这 6 名客人与其他人握手 (任何客人都不与其配偶握手). 假设每位客人握手的次数不同 (有可能有人没有握手). 通过构建这种情况的图模型 (6 个客人和 Megabucks 先生对应于 7 个顶点), 确定 Megabucks 先生必须握多少次手. (*提示*: 首先确定 6 名客人的不同握手次数.)

21. 可以用完全有向图来表示循环赛, 其顶点是竞赛选手, 而如果 a 击败 b, 则有边 $(a^{\rightarrow}, b)$. 每名选手与其他每名选手比赛一次. 一个顶点 (选手) 的得分就是它的出度数 (获胜的次数). 证明: 如果顶点 x 是循环赛各顶点中有最大得分的顶点, 那么对于任意其他顶点 y, 或者有边 $(x^{\rightarrow}, y)$, 或者存在某个顶点 w, 使得存在边 $(x^{\rightarrow}, w)$ 和 $(w^{\rightarrow}, y)$.

22. 假设循环赛图 (练习 21) 没有有向回路. 如下定义顶点 (选手) 的等级. 没有外向边的顶点的等级是 0. 一般情况下, 一个顶点有等级 k, 如果它有一条指向等级为 $k-1$ 的顶点的边, 且它的所有外向边都指向更小等级 ($\leqslant k-1$) 的顶点. 证明: 没有有向回路的有向完全图总是有这样的等级且每个顶点有不同的等级.

23. *迹* (trail) 是这样的顶点序列, 相继的顶点由不同的边连接 (边不能重复). 与路径不同, 在迹中, 顶点可以访问任意多次. *圈* (cycle) 是开始并结束于相同顶点的迹. 不含重复顶点的圈是回路.

(a) 证明: 可以使用从 x 到 y 的迹上的顶点的子集来做从 x 到 y 的路径.

(b) 证明或举出反例：如果 x 和 y 位于一条环上，那么它们一定位于一条回路上.

(c) 证明：可以把圈上的边划分成回路的集合.

(d) 证明：如果 C 是奇长度的圈，那么 C 的边的一个子集形成一个奇长度的回路.

24. 考虑平面上 (不同大小的) 圆的集合. 做一个圆图 (circle graph)，每个圆对应一个顶点，若圆相交则其对应的顶点连一条边 (一个圆包含另一个圆时没有边).

(a) 画出其圆图与 K_4 同构的一组圆.

(b) 画出其圆图是图 1.3 的一组圆 (不计边的方向).

(c) 画出其圆图与 $K_{3,3}$ 同构的一组圆.

25. 对于连通无向图 G，割集 (cut-set) S 是边的集合，它的删除将会断开 G，但 S 的任意真子集都不断开 G.

(a) 寻找图 1.22a 的最小割集.

(b) 证明：每个割集与任意回路都有偶数条公共边.

26. 有 n 个顶点和 $n+2$ 条边的图一定包含两个不含公共边的回路. 证明或举出反例.

27. 证明：如果 n 顶点图的边数大于 $1/2(n-1)(n-2)$，那么它一定是连通的. (提示：当包含两个连通分量，且每个连通分量都是完全子图时，非连通图包含最多可能的边.)

28. 证明：n 顶点图的边数大于 $1/4n^2$ 时，它一定不是二部图.

29. 假设 G 是不包含三角形的连通图，且 G 中每一对不相邻顶点正好有两个相同的邻居. 证明：G 中的每一个顶点都有相同的度. (提示：证明任意一对相邻顶点一定有相同的度.)

30. (著名的拉姆齐理论问题) 设 6 顶点完全图的每一条边都被染成红色或白色. 证明：总存在 3 条边组成的红色三角形或 3 条边组成的白色三角形.

31. (a) 寻找与其自身的补图同构的图.

(b) 证明：任意的自补图 [如问题 (a) 中的图] 一定有 $4k$ 或 $4k+1$ 个顶点，其中 k 是整数. (提示：利用 G 与 $\overline{G}$ 有相同边数的事实.)

32. 假设某个 7 顶点可平面图的每一条路径都有偶数条边 (0 条边、2 条边、4 条边等). 画出这个图. (提示：这是一个 "计策" 题.)

33. 证明：有向图没有有向回路，当且仅当可以把它的顶点标为 $x_1, x_2, x_3, \cdots, x_n$，使得所有边都形如 $(\overrightarrow{x_i, x_j})$，其中 $i < j$.

34. 设 $K_{m,n}$ 是一个完全二部图，且 $m > n$. $K_{m,n}$ 的最小边覆盖的大小是多少？最大独立集合的大小是多少？

35. 图 G 的线图 $L(G)$ (line graph) 是这样的图：以 G 的每条边为顶点，对应在 G 中有公共端顶点的两条边的顶点间有一条边.

(a) 画出图 1.6 中的左图的线图.

(b) 证明：$L(K_n)$ 中的每个顶点的度数是 $2(n-2)$.

(c) 寻找所有与自身的线图同构的图.

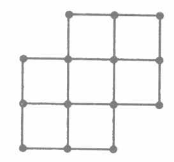
36. 证明：如果图 H 是某个图的线图 (参见练习 35)，那么可以把 H 的边分割成一组完全子图，使得 H 中的每个顶点正好在两个完全子图中.

37. 一个图的自同构 (automorphism) 是与它自身的顶点间的同构 (保持相邻关系的一一映射). 寻找下面各图的自同构.

(a) 图 1.1a.　　(b) 图 1.4.　　(c) 图 1.13.

38. (a) 证明：没有办法用 7 条边给下图中的 14 个顶点配对.

(b) 把问题 (a) 扩展成这样的一个问题：尝试使用 31 个骨牌覆盖除去了两个对顶角方格的 8×8 的棋盘上的 62 个方格.



39. 假设回路 C_1 和 C_2 有公共边 (但 C_1 不等于 C_2). 证明：$(C_1 \cup C_2) - (C_1 \cap C_2)$ 中的边构成一条回路 (或回路的集合).

40. 设图 G 有 $2n$ 个顶点且没有三角形, 证明 G 的边数不超过 n^2.

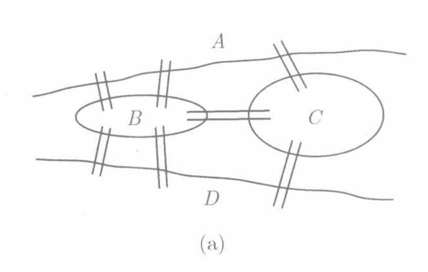
第 2 章　覆盖回路和图着色

2.1　欧　拉　圈

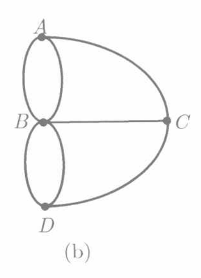
本章要研究两个有很多重要应用的图论概念. 一个概念是边序列, 它访问图中所有边一次或者所有顶点一次. 另一个概念是图着色, 我们曾经在 1.4 节的例 1 中提到过这个概念.

在某些应用中, 例如, 连接若干城市的高速公路网络, 允许若干条边连接相同顶点对是很自然的. 这样的扩展图称为*多重图* (multigraph). *多重图还允许形如* (a,a) 的循环边. 图 2.1b 给出了一个多重图.

在第 1 章的开始, 我们定义*路径* (path) $P = x_1 - x_2 - \cdots - x_n$ 为一个不同顶点的序列, 其中, P 中每一对相继顶点由一条边连接. 如果还存在边 (x_n, x_1), 则称这个序列为*回路* (circuit), 写作 $x_1 - x_2 - \cdots - x_n - x_1$. 与扩展图的概念一样, 现在要扩展路径和回路的定义, 其目的是为了允许有重复顶点. **迹** (trail) $T = x_1 - x_2 - \cdots - x_n$ 是一系列顶点 (有可能相同), 如同路径一样, 其中相继的顶点对由一条边连接. 然而, *迹中不含重复边*. 在图 2.1b 中, $A - B - D - B - C$ 是一个迹 (我们假设在这个迹上, 片段 $B - D$ 和 $D - B$ 使用了两个不同的边连接 B 和 D). 当还存在边 (x_n, x_1) 时, 这个顶点序列被称为**圈** (cycle), 写作 $x_1 - x_2 - \cdots x_n - x_1$.



(a)



(b)

图　2.1

例 1　柯尼斯堡桥

古老的普鲁士城柯尼斯堡位于 Pregel 河畔. 城中有两个小岛, 有七座桥把这两个小岛连接到岸边或相互连通, 如图 2.1a 所示. 城市居民喜欢围绕着这些桥在城市散步. 一些人似乎被这样的事实困扰: 没有人能指出经过每一座桥正好一次的路线. 他们就这个问题请教著名数学家欧拉. 欧拉解决了这个散步问题, 因而产生了图论并使柯尼斯堡七桥问题在数学教科书成为永恒的话题.

可以用一个多重图来对这个散步问题建模, 其中多重图的顶点是每一块陆地, 而边则是每一座桥. 参见图 2.1b. 要求的散步模式对应于我们现在所说的欧拉圈. **欧拉圈** (Euler cycle) 是包含图中所有边的圈 (并访问每一个顶点至少一次). 读者应该自己确认一下, 对于这个多重图, 不存在欧拉圈. □

带欧拉圈的多重图的每一个顶点的度都必须是偶数, 因为每当这个圈经过一个顶点时, 它使用两条边. 第二个明显性质是这个多重图必须是连通的. 欧拉证明这两个性质对于保证欧拉圈的存在也是充分的. 我们将对无向图证明这个定理. 首先, 在连通且顶点的度都为偶数的图中通过特定方法构建欧拉圈. 然后, 把我们的构建扩展到欧拉圈定理的一般证明.

例 2 构建欧拉圈

为图 2.2a 构建欧拉圈. 该图连通且顶点的度都为偶数.

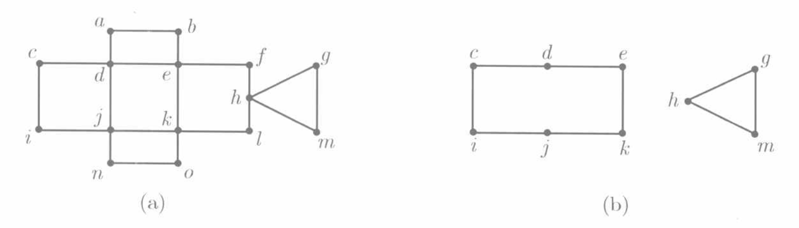


图 2.2

首先, 从顶点 a 开始尝试构建一个迹. 假设我们经过 $a-d-j-n-o-k-l-h-f-e-b-a$. 现在, 返回到 a. 注意, 度为偶数的性质意味着, 除了 a 之外, 总可以离开我们进入的任何顶点. 不存在度为 1 的顶点, 以及经过若干次访问之后度减少到 1 的顶点. 因此, 我们从 a 开始跟踪的任意迹都不会被迫在另一个顶点结束, 最终一定会返回到顶点 a, 从而形成一个圈; 对于本例的情况, 这个圈是回路 (没有顶点被访问两次).

接下来, 考虑剩余边的图, 如图 2.2b 所示. 它不再是连通的, 但是所有顶点的 (剩余) 度仍然是偶数 (删除这个圈使得每个顶点的度减少一个偶数). 剩余图的每一个连通部分都有一个欧拉圈: $d-c-i-j-k-e-d$ 和 $h-g-m-h$. 可以把这两个循环分别插入到原来循环的顶点 d 和 h 处, 从而产生欧拉圈 $a-d-c-i-j-k-e-d-j-n-o-k-l-h-g-m-h-f-e-b-a$. □

定理 1(欧拉, 1736) *无向多重图有欧拉圈, 当且仅当它是连通的且所有顶点的度都为偶数.*

证明 本证明扩展例 2 中的构建. 正如上面所提到的那样, 当欧拉圈存在时, 这个多重图一定是连通的且所有顶点的度都是偶数.

假设多重图 G 满足这两个条件. 任选一个顶点 a 并从 a 出发绘制一个迹. 如例 2 所示, 度为偶数的条件意味着我们不会被迫在某个其他顶点处停止, 所以最终

这个迹一定终止于 a (在最后被迫在 a 处停止之前, 它可能经过 a 若干次).

注意, 从开始时, 离开 a 之后, a 的 (剩余) 度为奇数. 这就是为什么我们最终被迫在 a 处停止的原因. 设 C 是这样生成的循环, 设 G' 是包含 G-C 的剩余边的多重图. 如例 2 所示, G' 可能不是连通的, 但是它的所有顶点的度仍然是偶数. 因为原始图是连通的, 所以 C 和 G' 一定有公共顶点, 否则不存在从 C 的顶点到 G' 的顶点的路径. 设 a' 是这样一个公共顶点. 现在, 构建从 a' 开始穿越 G' 的圈 C', 就如同在 G 中从 a 开始跟踪 C 一样. 同例 2 一样, 把 C' 在顶点 a' 处与 C 合并, 得到一个新的更大的圈 C'. 重复上述过程, 通过跟踪剩余边的图 G'' 中的圈, 并把这个圈合并到 C' 中得到圈 C''. 持续这一过程, 直到不存在剩余边. ◇

现在, 考虑欧拉圈在城市系统问题中的应用.

例 3　穿行街道的清扫车

假设图 2.3 中的实边图表示某个城区的街道清扫车从上午 10 点到上午 11 点要清扫的一些区域 (期间禁止在这些区域内停车). k 处的循环边表示一个圆形街道. 我们想要寻找每条实边清扫一次的行程, 即寻找一个实边的欧拉圈. 遗憾的是, 在这样的应用中, 要清扫的边所构成的图的所有顶点的度很少都是偶数, 而且通常也不是连通的. 在这样的情况下, 我们必须经过称为*空车返回边* (deadheading edges) 的其他一些边, 以得到想要的行程. 这就产生了一个新问题: 如何使空车返回的边数最少.

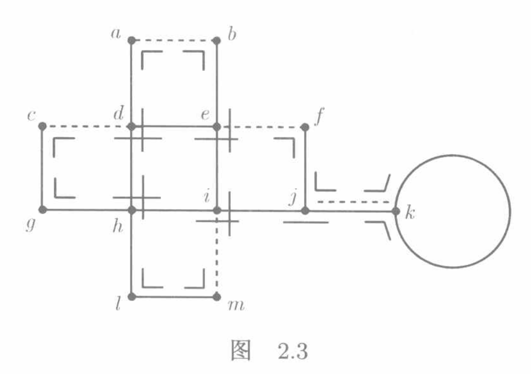
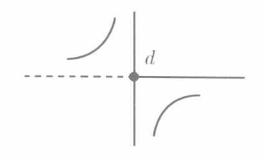


图　2.3

要求的行程是清扫边和空车返回边的多重图上的欧拉圈. 这是一个多重图, 因为在空车返回时有些边可能被重复. 根据定理, 这个多重图必须连通且所有顶点的度都是偶数. 因此, 为使空车返回边最少, 空车返回边应该是具有下面性质的最小额外边的集合, 即当这些边加入到原始图时, 所有顶点的度都是偶数且新的多重图是连通的. 假设图 2.3 中的虚线边是这样的额外边的最小集合.

一旦有了欧拉圈的多重图, 我们就需要面对另一个问题. 我们希望构建一个在拐角处拐弯数目最少的欧拉圈, 特别是 U 形拐弯 (掉头) 的数目最少. 在繁华的城市, 即便是右转弯也会妨碍交通. 我们可以依次考察每一个拐角 (顶点), 并配对这

个拐角处的边, 使得穿过这一拐角的行程中断次数最少. 对于图 2.3 中的多重图, 在访问拐角 d, e, h, i 时, 我们将直行穿过这些拐角; 我们将直行穿过 j 一次, 转弯一次 [在 (f, j) 和 (k, j) 间经过]; 我们必须在 k 处拐弯两次; 而在所有其他拐角处 (度为 2) 我们必须转弯. 在图 2.3 中, 每个拐角旁边的短线给出了这些边配对. 当然, 这样的边配对不大可能产生唯一的欧拉圈. 在图 2.3 中, 我们根据这些配对得到两个循环: $a-b-e-i-m-l-h-d-a$ 和 $c-d-e-f-j-k-k-j-i-h-g-c$. 现在, 在公共顶点 d 处, 我们把这两个圈断开, 并把它们结合到一起. 例如, 在顶点 d 处, 改变边配对为如右图形式, 其结果是除一个拐角外其他顶点都有最佳边配对的欧拉圈.



在拥有 200 条边的大型街道网络中, 当在每一个拐角处执行最小中断边配对时, 这个网络可以形成 3 到 4 个圈, 因此, 也只需改变少量的边配对. 在实践中, 我们还需要使用有向图, 因为街道清扫车在它们清扫的一侧不能逆行. □

有趣的是, 街道清扫问题给出欧拉圈定理的另一个证明.

定理的另一个证明 度为偶数的条件意味着我们能够配对每一个顶点处的边 (并把它们链接在一起). 这些链接边形成一组循环. 连通性条件意味着这些循环有公共顶点, 在这些公共顶点可以把它们连接起来形成单个圈, 即欧拉圈. ◇

这一证明不如我们原来的路径跟踪方法直观, 但是它更简单且更加实用.

我们通过把欧拉圈的概念扩展到欧拉迹来结束本节的内容. *欧拉迹* (Euler trail) 是包含图中所有边的迹 (而且访问每个顶点至少一次).

推论 *多重图有欧拉迹但没有欧拉圈, 当且仅当它是连通的且正好有两个度为奇数的顶点.*

证明 假设多重图 G 有欧拉迹, 但没有欧拉圈. 我们称 G 的欧拉迹为 T. 这时, T 的开始和结束顶点的度必须是奇数, 而所有其他顶点的度都是偶数 (根据欧拉圈的所有顶点的度为偶数的证明中相同的理由). 这个图还必须是连通的.

另一方面, 假设多重图 G 连通且正好有两个度为奇数的顶点 p 和 q. 在 G 中增添边 (p, q), 得到 G'. G' 连通且所有顶点的度都是偶数. 因此, 根据欧拉圈定理, G' 有欧拉圈, 称它为 C. 现在, 从 C 中删除边 (p, q). 这一删除把欧拉圈简变成包含 G 的所有边的欧拉迹. ◇

练 习 2.1

练习概况 前 4 道练习都是关于尝试构建欧拉圈的. 练习 5~8 和练习 11~14 给出欧拉圈定理的扩展及相关的其他问题. 其余问题包含构建模型和更深层的理论, 最后两道练习要求编写计算机程序.

1. **(a)** 构建图 1.8 的右图的欧拉圈.

(b) 对于删除边 (d, g) 的图 1.22a, 构建欧拉迹.

2. **(a)** 对于什么样的 n 值, n 顶点完全图 K_n 有欧拉圈?

(b) 存在有欧拉迹但没有欧拉圈的 K_n 吗?

(c) 对于什么样的 r 和 s 值, 完全二部图 $K_{r,s}$ 有欧拉圈?

3. 寻找有 7 个顶点的图 G, 使得 G 和它的补都有欧拉圈.

4. 画出图 1.4 时, 你最少需要抬起几次笔?

5. **(a)** 有欧拉圈的图能有桥 (桥就是删除后将使图不再连通的边) 吗? 证明或举出反例.

(b) 给出一个含桥的有 10 条边且有欧拉迹的图.

6. 对于有一个顶点 (没有边) 的图有欧拉圈的这一陈述给出一个正面说明和一个反面说明.

7. 假设在欧拉圈的定义中, 我们去掉似乎多余的要求: 欧拉圈访问每个顶点, 而只要求这个圈包含每条边. 现在, 证明欧拉圈定理为假. 画图说明为什么这个定理为假.

8. 证明: 如果连通图有 $2k$ 个度为奇数的顶点, 那么一定存在 k 条互不相交的迹[①], 它们合起来包含所有边.

9. 考虑要在一起进行汽车加速赛的 6 名选手 A, B, C, D, E, F. 下面矩阵中, 行 X、列 Y 项中的 1 表示 X 和 Y 要进行比赛. 当选手能够连着进行两场比赛 (而不是连着进行 3 场比赛) 时, 获得最佳效果. 能否设计一个比赛序列, 使得每场比赛 (除了最后一场比赛) 中的一名选手还参加其后的比赛 (但不连着进行 3 场比赛)? 如果有可能, 给出这个比赛序列 (选手配对); 如果不能, 解释为什么?

	A	B	C	D	E	F
A	—	1	0	1	1	0
B	1	—	1	1	0	1
C	0	1	—	0	1	0
D	1	1	0	—	1	1
E	1	0	1	1	—	1
F	0	1	0	1	1	—

10. 是否可以在一个 8×8 棋盘上移动马, 使得它做每一个可能的移动正好一次 (对于由马连接的两个方格, 当在这两个方格间沿任意方向移动时, 认为这两个方格间的移动完成)? [②]

11. 证明: 在例 3 中, 任意清扫问题中的空车返回边的最小集合, 是在度为奇数的不同顶点对间形成路径的一组边.

12. 证明欧拉圈定理的有向图版本: 有向多重图有有向欧拉圈, 当且仅当这个多重图 (不计方向时) 是连通的, 且每个顶点的入度等于出度.

(a) 模仿欧拉圈定理的证明, 陈述你对这个定理的证明.

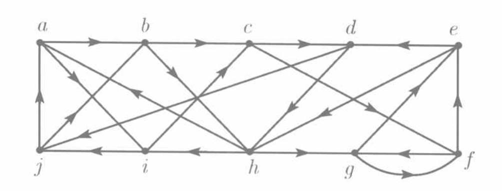
(b) 模仿欧拉圈定理的另一个证明, 陈述你对这个定理的证明.

(c) 为下面的图构建一个有向欧拉圈.

---

① 两条迹不相交的意思是它们没有共同边. —— 译者注

② 国际象棋中, 棋子放在棋盘的 8 × 8 方格内. 马每步棋先横走或直走一格, 然后再斜走一格 (每次斜走六个正方格子), 可以越子, 没有 "中国象棋" 中 "蹩马脚" 的限制. —— 编者注



13. 陈述并证明有向多重图版本的欧拉圈定理的推论.

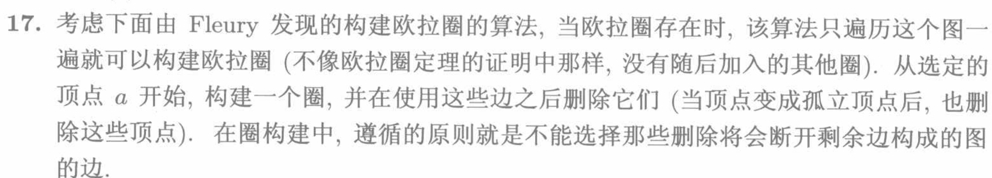
14. 有向图称为是**强连通的** (strongly connected), 如果从任意给定顶点到任意其他顶点间都存在有向路径. 证明: 如果有向图有有向欧拉圈, 那么它一定是强连通的.

15. 在右图中, 寻找符合下面条件的边的最小集合: 删除这些边将会产生一个欧拉圈. (**提示**: 需要一定的技巧.)

16. 图 G 的**线图** $L(G)$(line graph) 以 G 的边为顶点, 而其中两个顶点是相邻的当且仅当 G 中的对应边有公共端顶点.

(a) 证明: 如果 G 有欧拉圈, 那么 $L(G)$ 有欧拉圈.

(b) 寻找一个图 G, 它没有欧拉圈, 但其线图 $L(G)$ 有欧拉圈.



17. 考虑下面由 Fleury 发现的构建欧拉圈的算法, 当欧拉圈存在时, 该算法只遍历这个图一遍就可以构建欧拉圈 (不像欧拉圈定理的证明中那样, 没有随后加入的其他圈). 从选定的顶点 a 开始, 构建一个圈, 并在使用这些边之后删除它们 (当顶点变成孤立顶点后, 也删除这些顶点). 在圈构建中, 遵循的原则就是不能选择那些删除将会断开剩余边构成的图的边.

(a) 对于下面各图, 运用这一算法构建欧拉圈.

(i) 图 2.2a. **(ii)** 图 2.3 (包括空车返回边).

(b) 证明这一算法可行.

(c) 对于欧拉迹, 这一算法可行吗? 请解释.

18. 假设我们有代表双行街道网络的无向连通图.

(a) 证明: 在这个网络中总存在一个行程, 按这一行程一个人可以沿每条街道的每一侧驱车一次.

(b) 证明部分 (a) 中的行程可以通过下面的规则生成: 在任意交叉口处, 只有在从这个交叉口出去的其他街道都已用完的情况下, 才能经由到达这个交叉口的街道离开 (掉头).

19. 一组 8 位二进制数字 (0 或 1) 匀距离地放置在一个磁盘的边上. 我们要选择一些数字, 使得它们形成一个循环序列, 在这个序列中, 每一个长度为 3 的子序列都互不相同. 用有 4 个顶点的图对这个问题建模, 每个顶点代表一个两位二进制数子序列. 为每一个 3 位数字子序列做一条代表这个子序列的有向边, 其起点为代表这条边的子序列的前两个数字的顶点, 其终点是代表这个子序列的后两个数字的顶点.

(a) 构建这个图.

(b) 说明欧拉圈如何对应于所要求的 8 位圈序列 (对于此图, 存在欧拉圈).

(c) 使用这个图模型, 寻找这样的 8 位圈序列.

(d) 对于 4 位二进制序列, 重复这一问题.

20. 编写一个计算机程序, 当多重图中存在欧拉圈时, 该程序寻找一个欧拉圈:

(a) 使用欧拉圈定理的证明中的方法.

(b) 对有向图重复部分 (a).

(c) 使用欧拉圈定理的另一种证明中的方法.

21. 编写实现练习 17 中的算法的程序.

2.2 哈密顿回路

本节探讨访问图中每个顶点正好一次的**哈密顿回路** (Hamilton circuit) 和**路径** (path). 人们在探求诸如必须到若干家商店送货的卡车行程等这类运筹学的问题中形成了哈密顿回路的概念. 在这些应用中, 人们利用寻找到最小费用哈密顿回路来得到最有效的解决方案 (我们在 3.4 节讨论这一问题).

欧拉圈定理回答了确定一个图是否有欧拉圈的问题, 欧拉圈定理告诉我们, 一个图有欧拉圈当且仅当所有顶点的度都为偶数且这个图是连通的. 这样简单、美妙的答案在图论中非常少见 (这也可能是欧拉圈定理成为图论中第一个要证明的定理的原因). 本节, 我们回归图论的常态: 不存在确定一个任意的图是否有哈密顿回路或哈密顿路径的简单方法. 实际上, 在图中寻找哈密顿回路或路径是一个 NP 完全问题 (参见附录 A.5).

对于中等大小的图来说, 当哈密顿回路存在时, 通过观察来寻找它们通常不是太困难, 但是证明给定图不存在哈密顿回路则可能非常困难. 这样的证明通常需要使用与证明两个图不同构或一个图不是可平面图时所需的推理类似的推理. 本节的结尾给出在哈密顿回路存在性的研究中已经发展出来的理论样本. 这些定理给出保证一个图存在哈密顿回路的各种特殊条件. 大多数图不满足这些条件.

我们主要是要证明, 对于一些特殊的图, 不存在哈密顿回路. 这种不存在性问题需要大多数作为应用图论根本的系统逻辑分析. 为了证明不存在性, 必须从构建部分哈密顿回路开始, 然后证明这种构建一定总是失败的. 这类似于 1.4 节中的方法, 在那里, 我们通过构建一个回路, 然后把弦加入到已构建的模式中, 迫使两条边相交, 从而证明一个图不是可平面图.

我们的分析使用了用于构成哈密顿回路的边的集合必须满足的 3 条简单的法则. 这些法则的基本思想是: 任意哈密顿回路必须正好包含与每个顶点相连的两条边.

法则 1. 如果一个顶点 x 的度为 2, 那么与 x 相连的两条边必须是任意哈密顿回路的一部分.

法则 2. 没有真子回路, 即, 构建哈密顿回路的过程中, 不能形成不包含所有顶点的回路.

法则 3. 一旦需要哈密顿回路使用某个顶点 x 处的两条边, 那么就不能再考虑与顶点 x 相连的其他所有 (没有使用的) 边.

例 1　哈密顿回路的不存在性 I

证明图 2.4 没有哈密顿回路.

我们在顶点 a, b, d, e 运用规则 1. 为了说明必须使用这些顶点中每一个顶点处的两条边, 我们从一条边到另外一条边画一个小线段, 如图 2.4 所示. 现在, 显现出来两种类型的矛盾. 首先, 对顶点 a 和 d 使用规则 1 会迫使哈密顿回路使用 (a, c), (a, d) 和 (d, c) 形成一个三角形; 这违反规则 2. 其次, 对度为 2 的所有 4 个顶点使用规则 1 会迫使哈密顿回路包含多于两条与 c 相连的边. 这两个矛盾证明这个图没有哈密顿回路. □

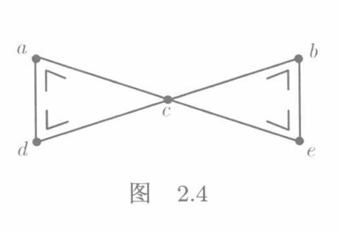


图　2.4

在接下来的两个例子中, 我们要把这 3 个规则用于下面这种顺序推理中. 当规则 3 要求我们删除某个顶点 q 处的边时, q 的被删除的一条边可能连接到度为 3 的顶点 r. 删除 (q,r) 后, r 的度为 2, 根据规则 1 的要求, 我们需要使用顶点 r 的剩余两条边. 当被迫删除某些边时, 就迫使我们使用其他某些边, 这又反过来迫使我们删除另外一些边, 以此类推.

例 2　哈密顿回路的不存在性 II

证明图 2.5 没有哈密顿回路.

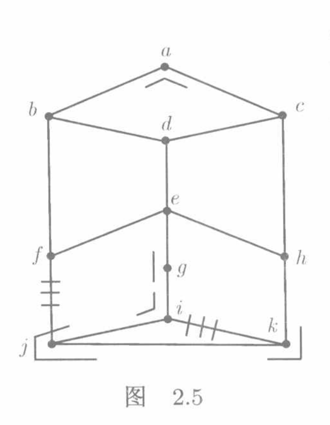


图　2.5

可以在顶点 a 和 g 处使用规则 1, 使得子路径 $b-a-c$ 和 $e-g-i$ 必须是任意哈密顿回路的一部分. 接下来考虑顶点 i, 我们已经知道 (g,i) 一定是哈密顿回路的一部分. 因为这个图是关于边 (i,j) 和 (i,k) 对称的, 我们选择哪条边来作为哈密顿回路中与 i 相连的边无关紧要. 假设选择边 (i,j) 来作为哈密顿回路的一条边. 如果我们使用 (i,j) 得到一个矛盾, 那么根据对称性, 使用 (i,k) 同样会得到一个矛盾. 可以把这种情况如下形象化: 如果在边 (i,j) 做上标志, 然后把画有这个图的纸张放在我们的眼前 (背对着我们), 并在镜子里看它, 这样就表现出我们是选择了边 (i,k); 从 (i,j) 开始的随后每一次移动都将产生从边 (i,k) 开始的镜像移动.

选择使用边 (i,j) 作为顶点 i 处的第二条边, 根据规则 3, 我们删除顶点 i 处的另一条边 (i,k). 参见图 2.5. 每一次运用一条规则, 我们都需要检查其他边和顶点受到了怎样的影响. 是否产生仅剩两个相连边的顶点, 从而可以运用规则 1 呢? 是否在某个顶点处出现了两条我们需要使用的边, 从而可以运用规则 3 呢? 还有, 是否某条边的使用会导致一个子回路, 从而可以运用规则 2 来删除它呢?

(i,k) 的删除把 k 的度减小到 2, 所以规则 1 要求我们使用与 k 相连的剩余两条边 (i,k) 和 (h,k). 边 (j,k) 是所使用的与 j 相连的第二条边. 于是, 根据规则 3, 必须删除 j 的另一条边 (f,j). 这条边的删除使得 j 的度减少到 2. 参见图 2.5. 所以我们必须使用 f 处的剩余两条边 (b,f) 和 (f,e). 边 (b,f) 是所使用的 b 处的第二条边, 所以必须删除 b 处的边 (b,d). 同样, 边 (f,e) 是所使用的 e 处的第二条边, 所以必须删除 e 的其他边 (e,d) 和 (e,h).

删除 (e,h) 迫使我们使用 h 处的边 (c,h). 使用 (c,h) 迫使我们删除 c 处的边 (c,d). 然而, 此时我们已经从哈密顿回路上删除了与 d 相连的所有边. 这一矛盾表明, G 不可能有哈密顿回路. □

注意, 这个图有哈密顿路径, 例如, $a-b-f-e-g-i-j-k-h-c-d$.

应该强调的是, 下面的推理路线对于证明一个图没有哈密顿回路没有什么用途: 从某个顶点开始, 构建一个访问后继顶点的路线, 并证明寻找经过所有顶点的回路的企图总是失败的. 即使是对于像图 2.5 那样简单的图, 也存在上百种可能成为哈密顿回路的开端的子路径, 这些子路径都需要我们去检查. 这里给出的方法比精确试错法所需的工作量少得多.

例 3 哈密顿回路的不存在性 III

证明图 2.6 没有哈密顿回路.

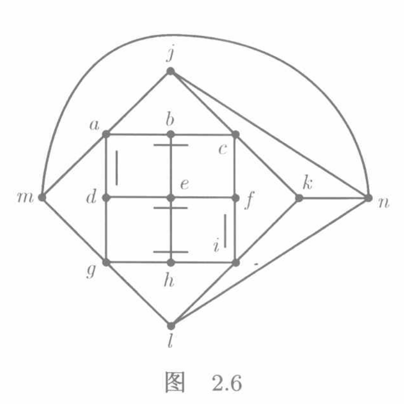


图 2.6

注意, 这个图关于垂直和水平方向对称 (尽管顶点 n 是在图的一边, 它的邻居是正方形对称的). 当不存在度为 2 的顶点时, 有时候要使用一些试错法的经验来寻找一个好顶点, 从这个顶点开始尝试着构建哈密顿回路. 我们寻找有这样性质的顶点: 一旦选择了这个顶点处的两条边, 那么法则 1 和法则 3 的使用迫使很多边被相继删除和加入. 顶点 e 是这样的一个顶点. 或者使用在 e 处相连且位于两侧的两条 (离开 180° 的) 边, 或者使用在 e 处相连且形成 90° 的两条边. 我们必须探查这两种情况来证明不可能存在哈密顿回路. (哈密顿回路通常有很多这样需要检测的子情况.)

情况 I 假设我们使用在 e 处相连且位于两侧的两条边. 由于对称性, 它们可以是来自于 d 和 f 的边, 也可以是来自于 b 和 h 的边. 假设选择 (d,e) 和 (e,f). 这时, 根据规则 3, 我们可以删除边 (e,b) 和 (e,h). 于是, 在 b 处和 h 处, 必须使用剩余的两条边, 得到子路径 $a-b-c$ 和 $g-h-i$. 现在, 在顶点 d 处, 可以使用边 (d,a), 或者边 (d,g). 这两种情况关于至此为这个回路所选择的边对称. 所以不失一般性, 可以选择 (d,a).

在顶点 f 处, 我们不能使用 (f,c), 否则产生子回路 $a-b-c-f-e-d-a$. 所以, 必须使用 (f,i). 参见图 2.6. 因为我们已经使用顶点 a 处和顶点 i 处的两条边, 根据规则 3, 这些顶点处的其他边必须被删除. 此时, 我们得到几个矛盾. 顶点 j,k,l,m 中每一个顶点的度都是 2. 但是它们每一个又都与 n 相连, 所以使用这些顶点中每一个顶点处的两条剩余边将使我们必须使用 n 处的 4 条边. 于是得出结论, 在情况 1 之下, 这个图没有哈密顿回路.

情况 II 假设我们使用在顶点 e 处相连且形成 90° 角的两条边. 根据对称性, 我们选择 4 对成 90° 角的边中的哪一对无关紧要. 假设选择 (b,e) 和 (d,e). 参见图 2.7. 这时, 根据规则 3, 我们要删除边 (e,f) 和 (e,h). 然后, 在顶点 f 和 h 处, 必须使用剩余的两条边, 得到子路径 $c-f-i$ 和 $g-h-i$. 参见图 2.7. 根据规则 3, 在顶点 i, 我们要删除边 (i,k) 和 (i,l). 根据规则 1, 必须使用顶点 k 和 l 处的剩余边, 形成子回路 $i-f-c-k-n-l-g-h-i$.

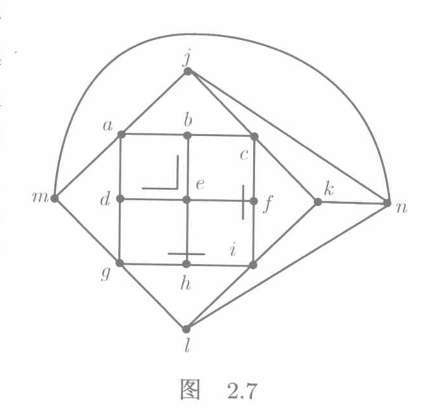


图 2.7

对于这两种情况, 我们都得到了矛盾, 这样就证明了这个图不存在哈密顿回路. □

现在给出关于哈密顿回路和路径存在的几个理论结果. (关于定理 1, 2, 3 的证明, 参见 [4].)

定理 1(Dirac, 1952) *有 n 个顶点的图 $(n>2)$ 有哈密顿回路, 如果每个顶点的度至少是 $n/2$.*

定理 2(Chvatal, 1972) *设 G 是有 n 个顶点的连通图, 且设把顶点标以 $x_1, x_2, \cdots, x_n$ 使得 $\deg(x_i) \leqslant \deg(x_{i+1})$. 如果对于每一个 $k \leqslant n/2$, 或者 $\deg(x_k) > k$ 或者 $\deg(x_{n-k}) \geqslant n-k$, 那么 G 有哈密顿回路.*

定理 3(Grinberg, 1968) *假设可平面图 G 有哈密顿回路. 设 G 是使用任意可平面图描述画出来的, 并设 r_i 是在这一描述下由 i 条边围成的哈密顿回路内部的区域数量. 设 r_i' 是由 i 条边围成的这个回路外侧的区域数量. 那么 r_i 和 r_i' 满足等式*

$$\sum_i (i-2)(r_i - r_i') = 0 \qquad (*)$$

正如我们可以使用 1.4 节推论中的可平面图不等式 $e \leqslant 3v-6$ 证明某些图不是可平面图一样, 可以使用定理 3 证明某些可平面图不存在哈密顿回路.

例 4 定理 3 的应用

证明图 2.8 中的可平面图没有哈密顿回路.

我们在图 2.8 的可平面图描述的每一个区域内标出了它的边界边的数量. 有 3

个有 4 条边的区域, 6 个有 6 条边的区域. 因此, 无论哈密顿回路画在那里 (如果存在的话), 我们都知道 $r_4+r_4'=3, r_6+r_6'=6$. 观察发现, 对于这个图, (*) 可以简化成:

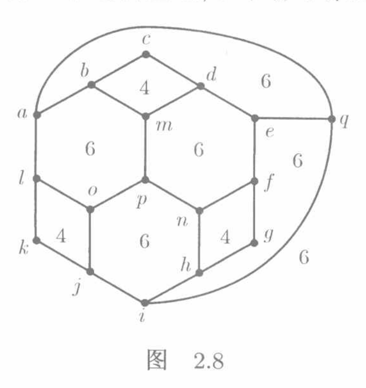


图　2.8

$$2(r_4-r_4')+4(r_6-r_6')=0$$

不能有 $r_6-r_6'=0$, 即, $r_6=r_6'=3$. 因为如果那样的话, (*) 将要求 $r_4=r_4'=0$ 或 $r_4=r_4'$, 这是不可能的, 其原因是 $r_4+r_4'=3$. 如果 $r_6-r_6'\neq 0$, 那么 $|r_6-r_6'|\geqslant 2$, 所以 $|4(r_6-r_6')|\geqslant 8$. 这不可能满足 (*), 因为即使 $r_4=3$, $r_4'=0$ (或者 $r_4=0$, $r_4'=3$), $|2(r_4-r_4')|\leqslant 6$. 因此, 对这个图, (*) 不可能为真, 所以这个图不存在哈密顿回路. □

下面给出一个有关有向图和哈密顿路径的定理. **循环赛** (tournament) 是通过为完全 (无向) 图的每一条边指定一个方向而得到的有向图.

定理 4　*每一个循环赛都有哈密顿路径.*

证明　使用归纳法证明这个定理. 对于一个 2 顶点循环赛, 显然存在有向哈密顿路径. 下面, 根据归纳法, 假设对于任意的 $n-1$ 顶点循环赛, $n\geqslant 3$, 有哈密顿回路, 我们证明 n 顶点循环赛 G 有有向哈密顿路径.

从 G 中删除顶点 z, 余下一个有 $n-1$ 个顶点的循环赛 G'. 根据归纳假设, G' 有哈密顿路径 $H=x_1-x_2-x_3\cdots x_{n-1}$. 如果 z 和 x_1 之间的边是 $(\vec{z},x_1)$, 那么可以把 z 加到 H 的前边, 得到一条 G 的哈密顿路径. 同样, 如果 z, x_{n-1} 之间的边是 $(\vec{x}_{n-1},z)$, 那么可以把 z 加到 H 的末端, 得到一条 G 的哈密顿路径. 所以假设从 H 的第一个顶点 x_1 出来的边指向 z, 而从 z 出来的边指向 H 的最后顶点 x_{n-1}. 这时, 对于 H 上的某个相继顶点对 x_{i-1}, x_i, 边的方向必须改变, 即我们有边 $(\vec{x}_{i-1},z)$ 和 $(\vec{z}, x_i)$. 把 z 插入到 H 中的 x_{i-1} 和 x_i 之间, 我们得到一条哈密顿路径 $x_1-x_2-\cdots x_{i-1}-z-x_i\cdots x_{n-1}$. ◇

在本节的最后, 我们举例说明哈密顿路径在编码理论中的应用.

例 5　格雷码

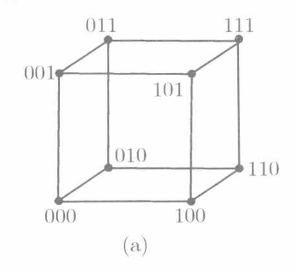
我们把太空飞船送往遥远的行星, 而它向地球传回图像, 这些图像都是以长长的数值序列的形式传回的, 每一个数都是这个图像中一个点的灰度值. 为简单起见, 假设这个灰度值的范围是 1 到 8. 这些数值实际上是以 0 和 1 的序列传回的. 一个直截了当的编码方案是用每个数值的二进制表示法表示它, 即 1 是 001, 2 是 010, 3 是 011, 等等, 最后 8 为 000.

然而, 有一种更好的编码方案, 称为*格雷码* (Gray code), 这一方案把数值编码成具有这样的性质: *相邻数值被编码成几乎相同的二进制序列, 它们只在一个位置上不同*. 例如, 一个格雷码的片段可能会把 4 编码为 010, 5 编码为 011, 6 编码为

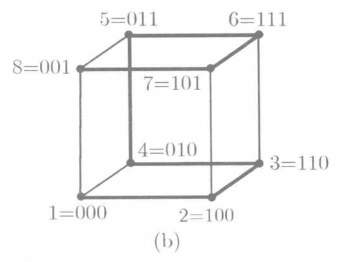
001. 这样编码的优点是, 如果因为“宇宙静电”所产生的误差导致序列中的某个二进制数字被地球上的接受站误读, 那么这个出错序列通常被解释成与正确灰度值几乎相同的灰度值. 例如, 在前面的格雷码片段中, 如果发送的是 011(5), 最后一位上出现了误差, 被接受成 010(4), 那么在灰度值中所产生的这种微小变化不会对图像产生严重影响. (当然, 有些错误会产生相当大的误差.)

在这样的背景下, 现在, 我们把寻找 8 个灰度值的格雷码问题转化成在一个图中寻找哈密顿回路的问题. 如下定义这个图: 每个顶点对应于一个 3 位二进制数字序列, 如果两个顶点仅在一个地方不同, 那么这两个顶点相邻. 图 2.9a 给出了这个图. 观察发现, 这个图是一个立方体 (可以把二进制数字序列想象成为三维空间中立方体的坐标). 还可以类似地画出 16 个顶点的 4 位二进制数字序列, 或者, 对于任意给定的 n, 我们可以画出 n 位二进制数字序列的 2^n 个顶点的图.

我们断言, 顶点 (二进制数字序列) 在这个图的一条哈密顿路径上出现的顺序产生一个格雷码. 即 1 被编码为给定哈密顿路径的第一个二进制数字序列 (顶点), 2 被编码为第二个二进制数字序列, 以此类推. 这个过程产生一个格雷码, 因为这条哈密顿路径上的相继顶点编码了相邻的灰度值, 而这些相继的顶点对应于仅在一个位置上不同的二进制数字序列. 图 2.9b 说明这个图中的一条哈密顿路径是如何产生一个格雷码的. □



(a)



(b)

图 2.9

每个顶点对应一个 n 位二进制数字序列, 且每条边连接了两个仅在一个位置上不同的二进制数字序列的图称为 n 维立方体, 或**超立方体** (hypercube). 图 2.9a 是一个 3 维立方体 (或标准立方体). n 维立方体有 2^n 个顶点, 每个顶点的度为 n. 它有这样的性质: 任意两个顶点间的最长路径是 n. 超立方体出现在超级计算机的设计中. 在一个有 2^n 个处理器的超级并行计算机中, 由于处理器太多, 我们无法直接连接每一对处理器. 取而代之, 一种超级计算机的设计方案是把处理器按超立方体图结构相互连接起来.

我们拓展哈密顿回路来解决本章补充材料中的四方柱问题.

练 习 2.2

练习概况 前 10 道练习是关于哈密顿路径和回路的存在性问题. 练习 9 和练习 10 介绍了其他一些对于证明不存在性问题的很有帮助的方法. 练习 15~19 是哈密顿回路的应用. 最后 5 道

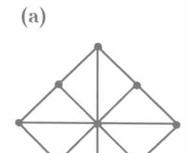
练习涉及一些理论问题.

1. **(a)** 画出一个有哈密顿回路但没有欧拉圈的图.

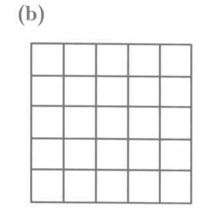
(b) 画出一个有欧拉圈但没有哈密顿回路的图.

2. 在下面各图中寻找哈密顿回路.

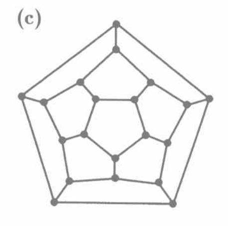
(a)



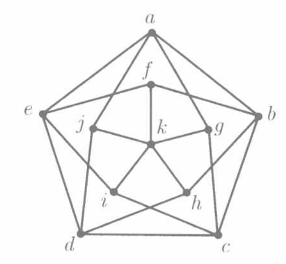
(b)



(c)

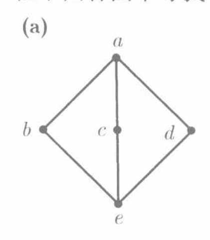


3. 在下图中寻找哈密顿回路.

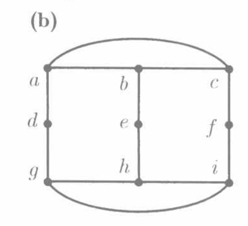


4. 在下面各图中寻找哈密顿路径, 并证明不存在哈密顿回路:

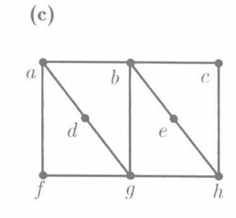
(a)



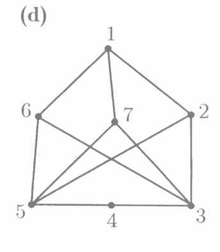
(b)



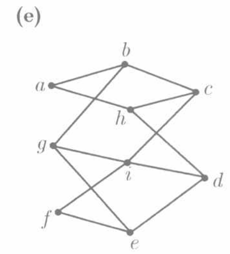
(c)



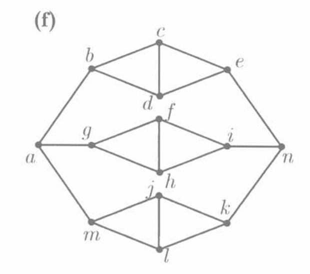
(d)



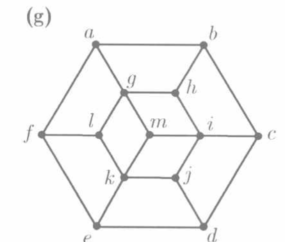
(e)



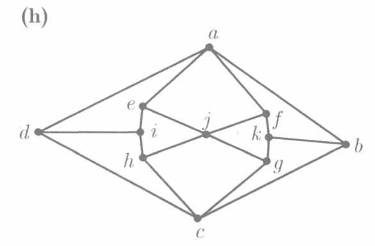
(f)



(g)

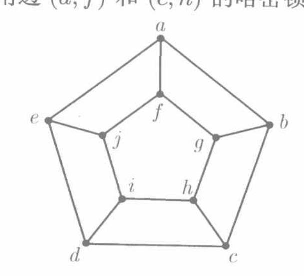


(h)



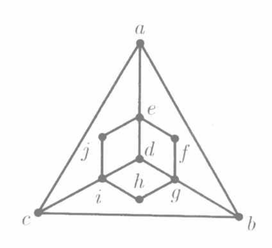
5. 在图 2.8 中寻找哈密顿路径, 并证明不存在哈密顿回路 (使用例 1~3 中的推理).

6. 证明: 在下图中, 不存在使用边 (a, f) 和 (c, h) 的哈密顿回路.

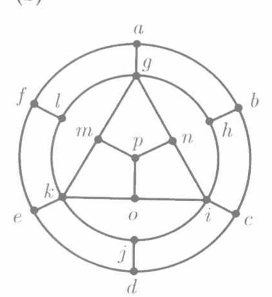


7. 证明：下列各图没有哈密顿回路：

(a)



(b)



8. 使用定理 3 证明下面各个可平面图没有哈密顿回路：

(a) 练习 4(a); **(b)** 练习 4(b); **(c)** 练习 4(p).

9. 在 1.1 节的例 1 中, 我们定义一个图是*二部的*, 如果顶点可以被分成两个集合：为了方便, 不妨称它们为蓝顶点和红顶点, 使得每条边连接一个蓝顶点和一个红顶点.

(a) 证明：如果连通二部图有哈密顿回路, 那么红顶点的数目与蓝顶点的数目一定相等. 因此, 如果二部图有奇数个顶点, 那么它没有哈密顿回路.

(b) 证明：如果连通二部图有哈密顿路径, 那么红顶点数目与蓝顶点数目的差至多是 1.

(c) 使用问题 (a) 证明下面各图没有哈密顿回路：

(i) 图 2.8; **(ii)** 练习 4(m); **(iii)** 练习 7(b).

10. 假设图 G 中 k 个顶点的集合 I 是一个独立集合, 即 I 中任意一对顶点都不相邻. 那么对于 I 中每一个顶点 x, 哈密顿回路一定不使用与 x 相连的 $\deg(x)-2$ 条边. 对于 I 中所有顶点求和, 我们有 $e'=\sum_{x\in I}(\deg(x)-2)=\left\{\sum_{x\in I}\deg(x)\right\}-2k$ 条边不能用于哈密顿回路中.

(a) 设 v 和 e 分别是 G 中的顶点数和边数. 证明：如果 $e-e'<v$, 则 G 没有哈密顿回路.

(b) 为什么要主张只有当 I 是不相邻顶点的集合时, 部分 (a) 中的观点才是正确的?

(c) 通过选择适当的集合 I, 使用问题 (a) 证明下面各图没有哈密顿回路：

(i) 图 2.6; **(ii)** 练习 4(p); **(iii)** 练习 7(b).

11. **(a)** 画一个 4 维超立体图.

(b) 使用问题 (a) 中的图寻找把数字 1 到 16 编码为 4 位二进制数字序列的格雷码.

12. 我们把连通图中两个顶点之间的距离定义为连接这两个顶点的最短路径上的边数. 这时, 图的*直径*(diameter) 定义为这个图中任意两个顶点之间的最大距离. 证明 4 维超立方体的直径为 4. 一般情况下, 证明 k 维超立方体的直径为 k. 注意：所有顶点的度为 k, 且有 2^k 个顶点的任意图的直径都不小于 k. k 维超立方体可以达到这个最小值.

13. 寻找没有哈密顿回路的连通立方体图 (所有顶点的度都为 3).

14. 不引用本节陈述的任何定理, 证明所有顶点的度都为 3 的任何 6 顶点无向图都有哈密顿回路.

15. 寻找马经过 8×8 的棋盘上所有方格恰好一次的路径. (提示：需要很强的技巧, 可以求助于网络.)

16. 假设一个班级有 25 名学生, 分座于 5 行 5 列的书桌旁. 老师要把每名学生调换到邻桌 (前面、后面、左边或右边) 来改变座次. 证明这样的调动是不可能的.

17. **(a)** 描述如何使用车构建一个包含 $n \times n$ 棋盘所有方格的回路, 其中 n 是偶数. 使用王如何构建呢?

(b) 设 n 是奇数, 重做问题 (a).

18. 考虑以 3×3×3 阵列组成的 27 个小立方体 (如在魔方中那样). 构建一个有 27 个顶点的关联图, 其中每个顶点对应一个小立方体, 如果两个顶点有相互接触的面 (而不仅仅是边), 则这两个顶点相邻. 这样的图有起始于中心立方体对应的顶点, 结束于对应于某个角落立方体的顶点的哈密顿回路吗?

19. **(a)** n 顶点完全图 K_n 中有多少条不同的哈密顿回路?

(b) 证明: 当 n 是不小于 3 的素数时, K_n 可以把它的边分割成 $1/2(n-1)$ 个不相交的哈密顿回路.

(c) 如果在一次会议中, 17 位教授围坐在一个圆桌就餐, 而且如果每天晚上每一位教授都与一对不同的教授邻座, 那么这个会议可以持续多少天?

20. **(a)** 如果图 G 有欧拉圈, 证明 G 的线图 (参见 2.1 节练习 16 对线图的定义) $L(G)$ 有哈密顿回路.

(b) 如果 G 有哈密顿回路, 证明 $L(G)$ 有哈密顿回路.

(c) 通过寻找反例, 证明问题 (a) 和 (b) 的逆为假.

21. 证明: 如果 G 不是完全图, 那么可以为 G 的边指定方向, 使得它不存在有向哈密顿路径.

22. 证明: 在一个 (前面的定理 4 定义的) 循环赛中, 总有可能给选手分出等级, 使得等级为 i 的选手打败等级为 $(i+1)$ 的选手. (*提示*: 运用定理 4.)

23. 证明: 把度 $\geqslant \frac{1}{2}n$ 的要求放宽到度 $\geqslant \frac{1}{2}(n-1)$ 时, 定理 1 不成立.

24. **(a)** 证明: 对于 $n \geqslant 3$, 有 n 个顶点且有至少 $\binom{n-1}{2}+2$ 条边的无向图一定有哈密顿回路.

(b) 证明: 当只有 $\binom{n-1}{2}+1$ 条边时, 问题 (a) 为假.

2.3 图 着 色

在 1.4 节的例 1 中, 我们介绍了地图着色问题: 给一个地图上的国家着色, 使得有公共边界的两个国家有不同的颜色. 证明任何地图只需用 4 种颜色着色的问题戏弄了数学家 100 年, 直到 1976 年 Appel 和 Haken 给出了一个计算机辅助证明. 再近一些, 图着色已经广泛运用于计算机科学、运筹学以及实验设计中的各种问题. 回想一下, 着色地图上的国家等同于着色一个图的顶点, 使得相邻的顶点得到不同的颜色, 而这个图以顶点表示国家, 有公共边界的两个国家对应的顶点间有一条边, 参见 1.4 节的例 1. 一般来说, 图 G 的**着色** (coloring) 就是给 G 的顶点分配颜色, 使得相邻顶点得到不同的颜色.

本节说明如何确定对给定图着色时所需的最少颜色数量. 这个最少颜色数量称为该图的**色数** (chromatic number). 我们还给出一些图着色的应用. 2.4 节将给出几个图着色的定理. 在图着色中, 有相同的颜色的顶点互不相邻 (任意一对都不

能用一条边连接起来). 在 1.1 节的例 5 中, 我们介绍了**独立集** (independent set) 这一术语, 它是指互不相邻的顶点的集合. 本节的后面, 我们将再次讨论这个例子.

对于有 15 或少于 15 个顶点的图来说, 通常不难推测它的色数. 为了严格证明一个图的色数是 k, 我们必须证明这个图不能用 $k-1$ 种颜色正确地着色. 证明一个图不是 $k-1$ 着色如同证明一个图没有哈密顿回路一样是一个 NP 完全问题 (参见附录 A.5). 在这种情况下, 我们的目标就是要证明, 能够为这个图所构建的任意 $(k-1)$ 着色都迫使两个相邻的顶点有相同的颜色.

例 1 简单图着色

寻找图 2.10 的色数.

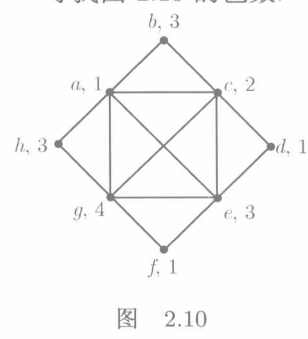


图 2.10

观察这个图中的内部有对角线的小正方形, 我们看到顶点 a, c, e, g 是彼此相邻的顶点, 即它们形成一个完全子图. 在正确的着色下, 它们需要不同的颜色, 总共需要 4 种颜色. 一旦 4 种颜色可用, 就很容易给剩余的顶点 b, d, f, h 着色. 这些顶点中的每一个顶点都只与其他两个顶点相邻, 所以这些顶点至多需要避开 4 种颜色中的 2 种. 我们使用数字 1, 2, 3, 4 来作为 4 种颜色的"名字". 于是, 图 2.10 给出这个图的一种可能的 4 着色.

对于这一问题, 显然这个图不可能是 3 着色的, 因为在由 a, c, e, g 所形成的完全子图中, 在 3 着色的情况下某个相邻的顶点对会有相同的颜色. 所以这个图的色数是 4. □

例 1 明确指出两个重要的规则. 第一, k 顶点完全子图需要 k 种颜色 [不可能是 $(k-1)$ 着色的]. 第二, 当构建某个图的 k 着色时, 我们可以忽视度小于 k 的所有顶点 (以及它们相关的边), 因为一旦其他顶点被着色, 总是至少存在一种可用的颜色 (任意相邻的顶点不使用的颜色) 来给每一个这样的顶点正确着色.

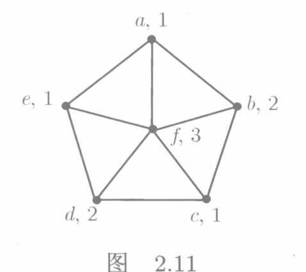


图 2.11

例 2 着色轮图

寻找图 2.11 的色数.

这种形式的图称为**轮图** (wheel). 这个图中最大的完全子图是三角形. 我们使用颜色 1, 2, 3 构建这个图的一个 3 着色. 先从着色三角形的顶点开始. 假设我们选择三角形 a, b, f: 让 a 使用颜色 1, b 使用颜色 2, f 使用 3(颜色的顺序是随机的). 参见图 2.11. 因为 c 分别与颜色 2 和 3 的顶点的 b 和 f 相邻, 所以只能让 c 使用颜色 1. 同样, 只能让 d 使用颜色 2, 然后只能让 e 使用颜色 1. 然而, 此时相邻的顶点 a 和 e 有相同的颜色 1. 因此, 这个图是不能 3 着色的. 另一方面, 如果使用第 4 种颜色给 e 着色, 则产生一个正确的着色. 所以这个图的色数是 4. □

观察发现, 在例 2 中, 如果没有顶点 e, 则顶点 d 与 a 相邻, 这时 3 种颜色是可行的. 一般情况下, 有偶数个辐条的轮图是 3 着色图, 而有奇数个辐条的轮图则需要 4 种颜色.

说明例 2 不可能有 3 着色的关键, 在于一系列被迫着色的顶点 c, d, e. 一般来说, 当试图构建一个 k 着色图时, 理想的做法是从对 k 顶点完全子图进行 k 着色开始, 然后不断地寻找与用前面 $k-1$ 种不同颜色着色的顶点相邻的未着色顶点, 这样迫使为这个顶点选择第 k 种颜色.

下面的例子涉及顶点颜色不是被迫着色的图.

例 3 非被迫着色

寻找图 2.12 的色数.

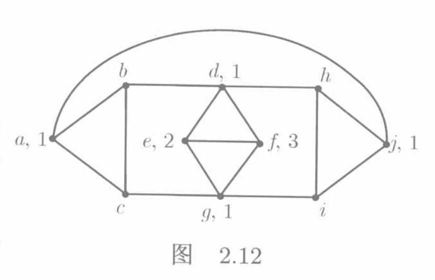


图 2.12

这个图的最大完全子图还是三角形, 所以我们想尝试着构建 3 着色. 仅存两个三角形 (d, e, f) 和 (e, f, g), 它们的着色将迫使其他顶点着色. 假设我们着色 d 为 1, e 为 2, f 为 3. 那么 g 被迫着色为 1. 此时, 我们陷入困境, 因为再没有未着色顶点与上面两种颜色相邻.

观察发现, b 和 c 都与颜色 1 的顶点相邻, 而且彼此相邻. 因此, b 和 c 中的一个必须是颜色 2, 而另一个是颜色 3. 因为这个图是对称的, 所以可以假设 b 是 2, 而 c 是 3. 重要的是 b 和 c 中的一个是颜色 2, 而另一个是颜色 3, 因为这样才迫使 a 是颜色 1. 同样, h 和 i 将有颜色 2 和 3, 迫使 j 是颜色 1. 但是相邻的顶点对 a 和 j 都有颜色 1. 因此, 这个图不可能是 3 着色的. 给 a 或 j 着色第 4 种颜色将产生一个正确的 4 着色, 所以这个图的色数是 4. □

现在研究图着色的某些应用. 我们从 1.1 节的例 5 开始, 这个例子是关于调度委员会会议时间的问题.

例 4 委员会会议安排

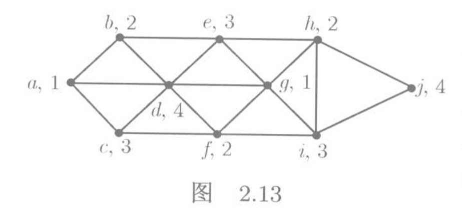


图 2.13

州立法机关有很多委员会, 这些委员会每周要举行为时 1 个小时的会议. 有人想要策划一个委员会会议时间表, 使得会议时间总数最少, 而且又使有交叉成员的两个委员会不在相同时间举行会议. 我们要说明这是一个图着色问题.

有关委员会的关键信息是哪些委员会有交叉成员. 我们创建这样一个图, 顶点对应于委员会, 如果两个委员会有交叉成员, 则这两个委员会所对应的顶点之间有一条边. 为了更具体一些, 假设图 2.13 表示 10 个立法委员会的成员交叉关系. 我们必须安排这些顶点, 使得 (有交叉成员的) 相邻顶点得到不同的颜色. 这个图的着色严格执行了这种 "调度" 形式. 颜色代表不同的会议时间. 图 2.13 的色数是 4(如

图 2.13 所示的最小着色). 因此, 4 个小时足以安排没有冲突的委员会会议. □

例 5 垃圾车调度

现在, 我们考虑一个复杂的优化问题, 在这个问题中, 图着色仅扮演次要的角色. 有一个每周必须被服务 (访问) k_i 次的场所 S_i ($1 \leqslant i \leqslant 6$) 的集合. 我们寻找在一周中卡车一天的最小行程集合, 使得场所 S_i 被访问 k_i 次. 另外, 我们要求这些行程能够以某种方式在一周的 6 天 (星期天除外) 之间分配, 使得在一天内任何场所都不能被访问两次. 这是一个相当困难的问题, 是无法精确地解决的.

在本书作者和一些同事为纽约市环境保护局分析这一类型的问题 (包括垃圾回收) 时, 我们使用了一个算法, 这一算法先从一个非高效的行程集合开始, 随后尝试着改进这个行程集合. 假设我们有一个简化的情况, 即一周有 3 个工作日, 并已经生成图 2.14a 所示的行程集合. 可以在这 3 个工作日分配这 6 个行程, 使得在同一天没有场所被访问两次吗? 描述这一限制的另一种方式是: 如果两个行程有一个共同的场所, 那么这些行程被分配给不同的工作日. 这种类型的限制是由着色模型来处理的.

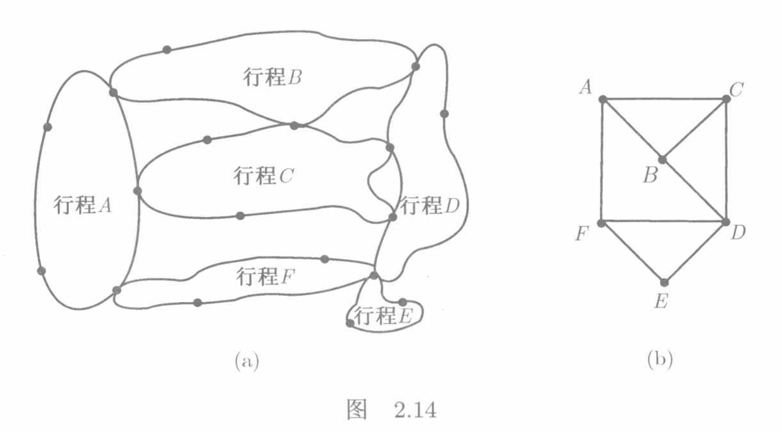


图 2.14

给定一个行程集合, 我们构建一个相关的行程图, 以每一个行程为一个顶点, 如果两个行程访问一个共同的场所, 则相应的两个顶点相邻, 即这两个顶点之间有一条边. 我们要给这个行程图的顶点分配颜色, 其中颜色代表不同的工作日. 在一般的 6 天问题中, 在 6 天之间分配行程就等同于 6 着色行程图. 在图 2.14b 的行程图中, 我们只是希望有一个 3 着色, 它确实存在. 如果优化算法近乎要合并图 2.14a 中的行程 A 和 F, 以得到一个更小的行程集合, 那么这样的优化动作需要被阻止 (需要去尝试其他的优化动作), 因为这样的合并所产生的行程图有一个由四顶点 A, B, C, D 构成的完全图, 这将需要 4 种颜色 (天). □

在本节的最后, 我们给出一个边着色的例子. **边着色** (edge coloring) 给每一条边分配一种颜色使得有公共端顶点的两条边有不同的颜色, 即与一个顶点相连的边

有不同的颜色.

例 6 循环赛赛程安排

在一个循环赛中, n 名选手中的每名选手都要与其余每名选手比赛一次. 循环赛的赛程安排是要列出每天进行哪些场次的比赛. 假设每名选手每天最多只能进行一场比赛.

(a) 把循环赛中的调度问题建模为边着色问题.

(b) 寻找有 4 名选手参加的循环赛的最小调度 (有最少天数).

(c) 寻找有 5 名选手参加的循环赛的最小调度 (有最少天数).

(a) 我们作一个图, 其中每名选手对应一个顶点, 一对选手之间有比赛则在对应顶点间有一条边. 这个图将是 n 顶点完全图 (每对顶点间都有边). 设循环赛的天数为颜色. 于是, 给循环赛中的每一场比赛分配日期使得没有选手在同一天比赛两场, 就等同于给 K_n 进行边着色, 使得与每个顶点相连的两条边没有相同的颜色.

(b) 我们需要寻找 K_4 的最小边着色. 给顶点 (选手) 命名为 a, b, c, d. 考虑 a 处的边. 设 a 在日期 1 与 b 比赛, 在日期 2 与 c 比赛, 在日期 3 与 d 比赛. 然后在每一天, 让不与 a 比赛的两个选手之间比赛. 用边着色来说明的话, 就是 (a, b) 与 (c, d) 有颜色 1, (a, c) 和 (b, d) 有颜色 2, (a, d) 和 (b, c) 有颜色 3.

(c) 我们需要寻找 K_5 的最小边着色. 给顶点 (选手) 命名为 a, b, c, d, e. 注意, 一天最多有两场比赛. 用边着色来说明的话, 就是对于每一种颜色, 最多有两条边有该颜色. 因为 K_5 有 10 条边 (参见图 2.15), 所以对于这个边着色我们至少需要 5 种颜色. 寻找 K_5 的边的 5 着色需要一点技巧. 我们使用下面的几何策略. 如图 2.15 所示, 设给定 K_5 外围的五角形的每条边有不同的颜色. 在图 2.15 中, 我们给边 (a, b) 颜色 1, 给边 (b, c) 颜色 2, 给边 (c, d) 颜色 3, 给边 (d, e) 颜色 4, 给边 (e, a) 颜色 5. 接下来, 对于内部呈星型的 5 条弦边的每条边, 我们指定和外面与它平行的边相同的颜色. 例如, 弦 (a, c) 平行于外面的边 (d, e), 所以给边 (a, c) 赋颜色 4. 因为平行边显然没有公共的端顶点, 这是一个正确的边着色.

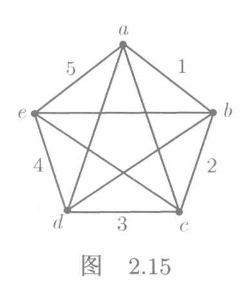


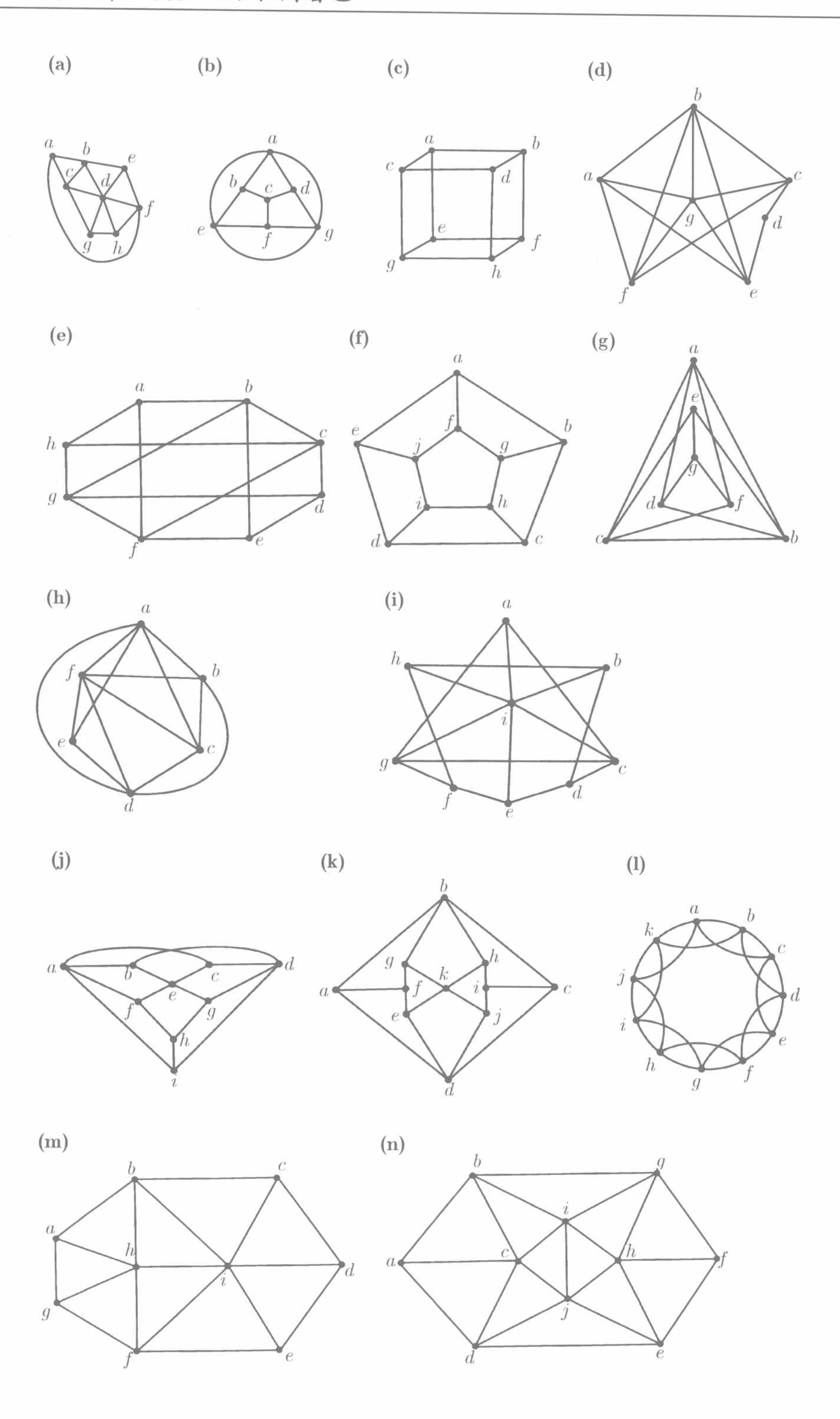
图 2.15

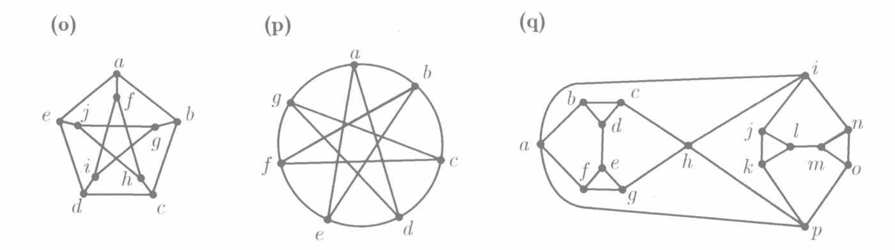
□

练 习 2.3

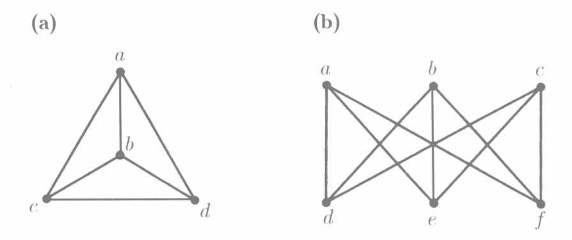
练习概况 练习 1~6 涉及寻找最小顶点着色以及相关的问题. 练习 7~9 需要地图和几何排列的最小着色. 练习 10~15 是着色建模问题.

1. 寻找下面各图的色数. 详细说明更少的颜色不足以着色.


(a)
a
b
e
c
d
f
g
h
(b)
a
b
c
d
e
f
g
(c)
a
b
c
d
e
f
g
h
(d)
b
a
c
g
d
f
e
(e)
a
b
h
c
g
d
f
e
(f)
a
e
f
b
j
g
i
h
d
c
(g)
a
e
g
d
f
c
b
(h)
a
f
b
e
c
d
(i)
a
h
b
i
g
c
f
e
d
(j)
a
b
c
d
e
f
g
h
i
(k)
b
g
h
k
a
f
i
c
e
j
d
(l)
a
b
k
c
j
d
i
e
h
f
g
(m)
b
c
a
h
i
d
g
f
e
(n)
b
g
i
a
c
h
f
j
d
e



2. 寻找下面各图的最小边着色 (给各边着色, 使得有公共端顶点的边得到不同的颜色).



(c) 练习 1(a). **(d)** 练习 1(b). **(e)** 练习 1(c).

3. 图 G 是色临界的 (color critical), 如果删除 G 的任意顶点都会减少色数. 练习 1 中的哪些图是色临界的?

4. 对于下面各图, 求所有在正确的着色下可以有相同颜色的 3 个或更多顶点的集合:

(a) 练习 1(a); **(b)** 练习 1(b).

5. 图 G 的顶点着色把 G (的顶点) 分割成若干个互不相邻顶点集合的族. 而 G 的补图 $\bar{G}$ 的顶点着色则把 G 分割成彼此相邻的顶点集合的族, 即, 完全子图 (的顶点) 的族. 寻找分割下面各图顶点的完全子图的最小集合:

(a) 练习 1(a); **(b)** 练习 1(b); **(c)** 练习 1(l).

6. **(a)** 把在例 6 中找到的 5 人循环赛的调度方案扩展到 6 人循环赛. (提示: 关键是 5 人循环赛中每天不进行比赛的那个人.)

(b) 把解决例 6 中的 5 人循环赛的几何方法运用到 7 人循环赛和 9 人循环赛, 寻找最小调度.

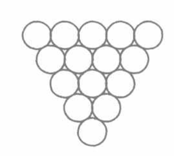
(c) 以问题 (a) 和 (b) 的结果为基础, 提出一个进行 n 人循环赛所需最少天数的一般公式. 注意, 答案依赖于 n 是奇数还是偶数.

7. 能够给美国地图上的 50 个州正确进行 3 着色吗? (注意, 我们认为只在一个角相遇的州不是相邻的州, 如科罗拉多州和亚利桑那州.)

8. 假设一张地图被画成 n 个相交圆. 证明可以给这个图中的各区域进行 2 着色.

(a) 用归纳方法证明.

(b) 根据包含一个区域的圆的数量分配颜色, 来给出证明.



9. 给右面这个排列成三角形的 15 个台球着色, 需要多少种颜色使相互接触的球有不同的颜色?

10. 一个动物园准备把动物散养在几个较大的空地区域, 而不是把它们各自关在笼子里. 如果两种不同动物不能和平地生活在一起 (例如, 虎和鹿不能生活在一起, 因为虎要吃掉鹿), 那

么必须把它们放在不同的空地上. 这个动物园想要确定给这些动物完全地安家所需的最少空地数. 把分配动物到最少空地的问题建模成图着色问题. 什么是顶点, 什么是边, 什么是颜色?

11. 应用数学系正在排下一学期的课表. 每名学生已经决定了自己要选修的若干应用数学课程. 必须排好课程, 使得每名学生能够没有冲突地上需要上的课程. 把这一课表安排问题建模成图着色问题. 什么是顶点, 什么是边, 什么是颜色?

12. **(a)** 若干观测站要做一组关于太阳的实验. 每项实验在一年的给定日期开始, 并在给定日期结束 (每项实验要重复若干年). 一个观测站一次只能进行一项实验. 问题是, 每年进行一组给定实验所需观测站的最小数目是多少? 把这一调度问题建模成图着色问题.

(b) 假设实验 A 从 9 月 2 日开始到次年 1 月 3 日结束, 实验 B 从 10 月 15 日开始到次年 3 月 10 日结束, 实验 C 从 11 月 20 日开始到次年 2 月 17 日结束, 实验 D 从 1 月 23 日开始到 5 月 30 日结束, 实验 E 从 4 月 4 日开始到 7 月 28 日结束, 实验 F 从 4 月 30 日开始到 7 月 28 日结束, 实验 G 从 6 月 24 日开始到 9 月 30 日结束. 画出相关图, 并寻找最小着色 (证明更少的颜色不足以正确着色).

13. 一个港口一周内要来 12 艘游轮. 必须给这些船分配 5 个码头中的一个 (一个码头一次只能有一艘游轮停靠). 问题是, 对于给定的游轮观光计划, 船到码头的分配是否可能. 描述如何建立这样的分配问题的图着色模型: 顶点是什么, 边是什么, 颜色是什么?

14. 宴会中心有 8 个包间. 每场宴会需要用其中的某几间. 假设在给定的一周内 (7 天) 要安排 12 场晚宴. 在同一个晚间安排的两场宴会必须使用不同的房间. 为这一调度问题建模并将其重述为图着色问题.

15. 对于图 2.14a, 考虑下面的行程对, 它们可以合并而不违反例 5 的行程图的 3 着色要求吗?
(a) 行程 D 和 E.　　**(b)** 行程 C 和 D.

16. 考虑表示若干足球队之间的比赛的图, 其中, 如果 A 队打败了 B 队, 则有从顶点 A 指向顶点 B 的有向边. 假设已知这个图*没有有向回路* (即, 没有 A 打败 B, B 打败 C, C 打败 A 的情况). 对于这个足球比赛图, 可以这样定义一组层次: 如果一个顶点没有指向外面的边 (这个队没有打败其他球队), 那么它位于层次 0; 如果一个顶点所有指向外面的边都是指向层次 0 的顶点, 那么这个顶点位于层次 1; 一般说来, 如果一个队所打败的球队的最高层次是 $k-1$, 那么它位于层次 k. 证明: 每个顶点的层次数是顶点的一个正确"着色".

2.4　着色定理

2.3 节研究了寻找图的最小着色的策略, 并给出了可以建模为图着色问题的一些应用. 本节给出几个关于图着色的定理.

我们先给出多边形的角着色定理, 然后再利用这个定理得到计算几何中一个有趣问题的简单解. 把多边形看成是由边被画成直线的单一回路组成的可平面图. 这个多边形不必是凸多边形. 参见图 2.16a. 提到*多边形的三角剖分* (triangulation of a polygon), 我们指的是在多边形顶点对之间加入一组直线弦的过程, 使这个图的所有内部区域都以三角形为界 (这些弦不能彼此相交, 也不能相交于多边形的侧边). 图 2.16b 给出了图 2.16a 的一种可能的三角剖分.

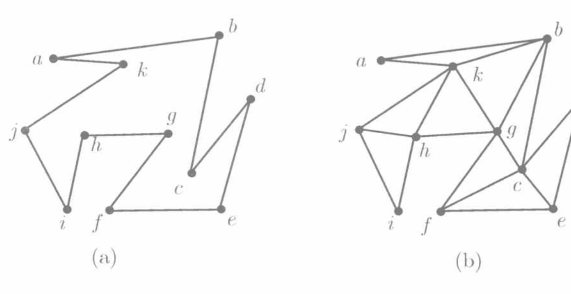


图 2.16

定理 1 *多边形的三角剖分是可以 3 着色的.*

证明 对多边形的边数 n 进行归纳证明. 当 $n=3$ 时, 给每一个角以不同的颜色. 对于任意的 $n \geqslant 4$, 假设边界边小于 n 的任意多边形的三角剖分都是可以 3 着色的. 现在考虑有 n 条边界边的多边形的三角剖分 T.

选择一个弦边 e, 例如图 2.17a 中的弦 (g,k). 我们注意到, 因为 $n \geqslant 4$, 所以 T 至少有一条弦边. 弦边 e 把 T 分割成两个更小多边形的三角剖分, 如图 2.17b 所示, 根据归纳假设, 每一个更小多边形的三角剖分都可以 3 着色. 把这两个子图的 3 着色合并成原始多边形的三角剖分的 3 着色, 其方法是在两个子图中选择使弦 e 的两个端顶点在每个子图中颜色相同的颜色名字. 在图 2.17b 中, 这就意味着使得 k 的颜色在两个子图中相同, 并使 g 的颜色相同. ◇

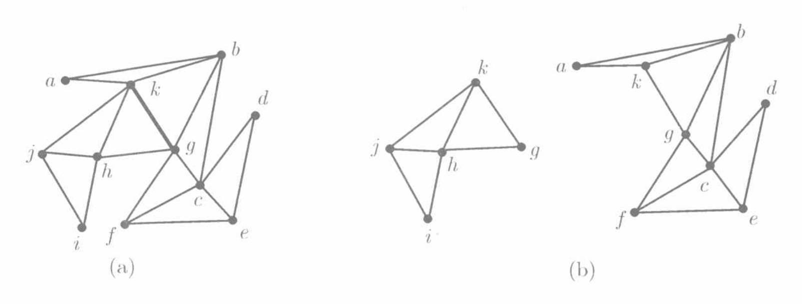


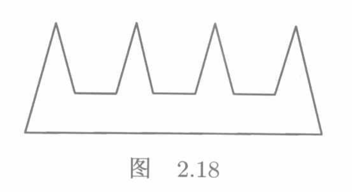
图 2.17

在实际应用中, 很容易生成多边形三角剖分的 3 着色. 而且, 这种 3 着色是唯一的. 细节可以参见练习 7. 现在, 我们给出定理 1 对表面上与着色没有关系的一个问题的有趣应用.

画廊问题要求有 n 面墙的画廊所需保安的最少人数. 假设这些墙形成一个多边形. 保安需要能够直接看到这些墙面上的每一点. 假设拐角处的保安能够看到交汇在该拐角的两面墙. 表达式 $\lfloor r \rfloor$ 表示小于等于 r 的最大整数. 关于画廊问题的更多信息参见 O'Rourke[5].

推论 (Fisk, 1978) *有 n 面墙的画廊问题至多需要 $\lfloor n/3 \rfloor$ 名保安.*

证明 做由画廊墙壁形成的多边形的三角剖分. 观察发现, 位于一个三角形任意角处的保安可以使这个三角形的所有边都在监控之下. 现在, 我们得到这个划分的一个 3 着色. 注意, 对于每一种颜色, 每一个三角形都有一个这种颜色的角. 取其中一种颜色, 比如说 "红色", 并在每一个被着色成红色的角落安置一名保安. 这样就在每一个三角形的一个角落处安置了一名保安. 因此, 所有三角形的各条边, 特别是画廊的各个墙面, 都在监控之下. 有 n 面墙的多边形有 n 个角. 如果有 n 个角和 3 种颜色, 那么存在一种颜色, 这一颜色被用于 $\lfloor n/3 \rfloor$ 或更少个角落处. ◇


图 2.18

这一界限值是最好的可能. 例如, 图 2.18 给出一个有 12 个角的多边形, 它需要 4 名保安.

现在, 我们给出 3 个代表性的着色定理. 我们只证明最后一个定理, 即可平面图的 5 着色定理. 对于定理 2 和定理 3 的证明, 参见 [4]. 注意, 1.3 节中的定理 2 是一个着色定理, 该定理说的是: 连通图是二部图, 当且仅当它的所有回路都有偶长度. 观察发现, 一个图是二部图与它可 2 着色是一样的. 因此, 一个图可 2 着色, 当且仅当它的所有回路都有偶长度. (可以把图连通的条件去掉, 因为如果每一个连通分量可 2 着色, 那么整个图可 2 着色.) 设 $\chi(G)$ 表示是图 G 的色数.

定理 2(Brooks, 1941) *如果图 G 不是奇回路或完全图, 那么 $\chi(G) \leqslant d$, 其中 d 是 G 的顶点的最大度数.*

对于大多数图来说, 这个最大度数不是 $\chi(G)$ 的好上界. 2.3 节中的例子中的 $\chi(G)$ 都与最大完全子图的大小有密切关系. 因此, 似乎很自然地想到, 应该可以用最大完全子图的大小来刻画 $\chi(G)$ 的好上界. 然而, 对于任意正整数 k, 存在不含三角形的图 G, 使得 $\chi(G) = k$, 其细节参见练习 18.

除着色顶点外, 还可以着色边, 使得有公共端顶点的边得到不同的颜色. 可以使用度来得到图的边色数的非常好的边界. 与给定顶点相连的所有边必须有不同的颜色, 所以图的顶点最大度数是边色数的下界. 更有甚者, 可以证明:

定理 3(Vizing, 1964) *如果图 G 的顶点的最大度数是 d, 那么 G 的边色数等于 d 或 $d+1$.*

最后, 我们想要给出并证明一个关于着色可平面图的定理. 如早前所述, 1976 年, Appel 和 Haken 证明了所有可平面图可以 4 着色 (参见文献 [1]). 但是他们的证明非常之长, 而且需要对 1955 种情况进行长达数千小时的计算机分析, 每一种情况又需要许多页的分析. 我们描述并证明一个更简单的 "次优" 定理.

定理 4 *所有可平面图都可以 5 着色.*

证明 只需考虑连通可平面图, 因为可以通过 5 着色每一个连通分量来着色整个非连通图. 证明中的关键步骤是使用 1.4 节练习 18 中证明的关于可平面图的事实: 任何连通可平面图都有一个度至多是 5 的顶点. 我们对顶点个数施用归纳法来证

明这个定理. 显然, 一个顶点的图是可以 5 着色的.

接下来, 假设所有 $n-1$ 个顶点的连通可平面图是可以 5 着色的 ($n \geqslant 2$). 我们证明有 n 个顶点的连通可平面图 G 可以 5 着色. 正如上面所述, G 有一个度至多为 5 的顶点 x. 从 G 中删除顶点 x, 我们得到一个有 $n-1$ 个顶点的图, 根据归纳假设这个图是可以 5 着色的. 现在, 把顶点 x 连回到上面 $n-1$ 个顶点的子图中, 尝试给顶点 x 正确着色. 如果 x 的度小于等于 4, 那么可以简单地给 x 分配一种与它的邻居不同的颜色. 如果 x 的度是 5, 但是有两个邻居有相同颜色, 那么同样可以给分配与它的邻居不同的颜色. 因此, 只剩下考虑 x 有 5 个相邻顶点且每一个相邻顶点都有不同的颜色的情况. 参见图 2.19, 在这个图中, 我们在 G 的某个可平面画法下, 按着顺时针方向给 x 的邻居标号为 a, b, c, d, e. 设它们的颜色是图 2.19 中所示的数字.

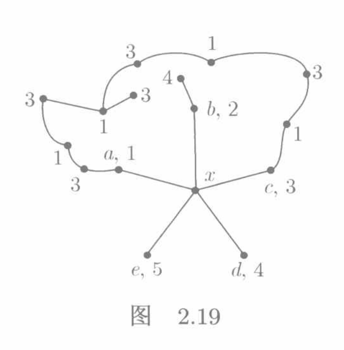


图 2.19

首先, 考虑从 a 出发的所有这样的路径, 这个路径的各顶点被着色成 1 和 3. 参见图 2.19. 假设从 a 到 c 没有由颜色 1 和颜色 3 的顶点组成的路径. 那么可以把 a 的颜色从 1 变成 3, 把 a 的被着色为 3 的邻居的颜色变为 1, 沿着由 a 出发的所有由颜色 1 和 3 的顶点组成的路径一直这样做下去. 这样的 1-3 互换不会影响到 c, 因为从 a 到 c 没有由颜色为 1 和 3 的顶点组成的路径. 经过了从 a 开始的 1-3 互换之后, a 和 c 都有颜色 3, 所以可以给 x 正确地着色为 1.

另一方面, 如果从 a 到 c 存在一条由颜色为 1 和 3 的顶点组成的路径, 那么考虑所有这样的从 b 出发的路径, 这条路径的顶点被着色为 2 和 4. 从 a 到 c 的颜色为 1 和 3 的顶点的路径加上边 (x, a) 和 (x, c) 组成一个回路, 这条回路阻断了从 b 到 d 存在颜色为 2 和 4 的顶点路径的可能性. 因此, 可以沿所有从 b 出发的颜色为 2 和 4 的顶点组成的路径实施 2-4 互换, 而不改变 d 的颜色. 经过这样的 2-4 互换之后, b 和 d 的颜色都是 4, 所以可以给 x 正确地着色为 2.

以上完成了证明的归纳部分, 即, 任意 n 顶点连通可平面图都可以 5 着色. ◇

本节的最后, 我们简单介绍一下色多项式. 图 G 的色多项式 $P_k(G)$ 给出了用 k 种颜色正确地着色 G 的方法数. 这一公式是一个关于 k 的多项式. 如果 k 很小以致于 G 不能只用 k 种颜色着色, 那么此时的 $P_k(G)$ 等于 0.

例 1　色多项式

下面各图的色多项式是什么:

(a) 路径 $P_6 = a-b-c-d-e-f$?

(b) 关于 5 顶点的完全图 K_5 (所有顶点彼此相邻)?

(c) 长度为 4 的回路 C_4?

(a) 对于顶点 a 有 k 种颜色选择. 对于 b 有 $k-1$ 种颜色选择 (除了 a 使用的颜色之外, 任意颜色都可以). 同样, 有 $k-1$ 种颜色可以用于着色后面的顶点 c, 然后是 d, e, f. 所以, $P_k(P_6) = k(k-1)^5$.

(b) 对于完全图, 每一个顶点必须有不同的颜色. 因此, $P_k(K_5) = k(k-1)(k-2)(k-3)(k-4)$, 因为对于要着色的第一个顶点, 有 k 种可以选择的颜色, 而所选的颜色之后就不能再使用了, 所以第二个要着色的顶点有 $k-1$ 种选择. 以此类推.

(c) 设回路 C_4 是 $a-b-c-d-a$. 我们根据 a 和 c 是否着色成相同的颜色来把 $P_k(C_4)$ 的计算分解成两种情况.

如果 a 和 c 有相同的颜色, 那么有 k 种颜色可以用于着色这两个顶点. 这时, b 和 d 都必须避开 a 和 c 的颜色, 因此有 $k-1$ 种颜色可以选择. 所以, 对于这一情况, C_4 的 k 着色数是 $k(k-1)^2$.

如果 a 和 c 有不同的颜色, 那么可以选出两种不同的颜色来着色 a 和 c, 共有 $k(k-1)$ 种选择. 现在, b 和 d 各有 $k-2$ 种颜色可以选择. 所以, 对于这一情况, C_4 的 k 着色总数是 $k(k-1)(k-2)^2$. 结合这两种情况, 我们得到 $P_k(C_4) = k(k-1)^2 + k(k-1)(k-2)^2$. □

可以扩展例 1 中的问题 (a) 和 (b) 的结果. 可以把 (a) 的结果扩展到没有回路的任意连通图. 这样的连通图称为树 (tree). 树是第 3 章的主题.

定理 5

(a) *如果 T 是有 n 个顶点的树, 那么 $P_k(T) = k(k-1)^{n-1}$.*

(b) *如果 K_n 是 n 顶点完全图, 那么 $P_k(K_n) = k(k-1)(k-2)\cdots(k-n+1)$.*

证明

(a) 在 T 中选择某个顶点 a. 可以选择 k 种颜色着色 a. 设 S_1 是与 a 相邻的所有顶点的集合. 注意 S_1 中两个顶点 x 和 y 之间没有边, 因为如果它们之间有边, 则 a, x, y 形成一个三角形, 即形成一个长度为 3 的回路. 这时, S_1 每一个顶点可以着色 a 的颜色之外的任意颜色: 每个顶点有 $k-1$ 种着色颜色. 现在设 S_2 是从 a 出发长度为 2(两条边) 的路径的末端的所有顶点. 为了避免形成回路, S_2 中任意两个顶点不相邻, 而每一个顶点又与 S_1 中唯一一个顶点相邻. 所以 S_2 中的每一个顶点还是有 $k-1$ 种着色颜色 (与其在 S_1 中的相邻顶点的颜色不同的任意颜色). 对于 S_3 和 S_4 等, 可以运用同样的讨论. 因此, 除了 a 之外, 每一个顶点都有 $k-1$ 种颜色选择.

(b) K_k 中的 n 个顶点必须有不同的颜色. 根据与例 1(b) 相同的推理可得其结果. ◇

注意, 如果一个图是非连通的, 那么它的色多项式是每一个连通分量的色多项式的乘积. 所以我们只需关心如何确定连通图的色多项式即可.

定理 5 是寻找任意连通图的色多项式的一个起点. 下面的分解定理说明如何反复把给定图的色多项简化成为两个缺失了更少边的更简单图的色多项式的和. 这

种化简最终导致完全图. 定理后面的推论给出了一种分解, 它产生有更少边的简单图, 并最终产生树.

定理 6 设 x, y 是图 G 的两个不相邻顶点. 设 $G_{+(x,y)}$ 是通过把边 (x, y) 加入到 G 中而得到的图. 设 $G_{x=y}$ 是通过把 G 中的顶点 x, y 粘合起来形成一个顶点 xy 而得到的图, 顶点 xy 相邻于 G 中与 x 或 y 相邻的任意顶点. 这时有

$$P_k(G) = P_k(G_{+(x,y)}) + P_k(G_{x=y}) \tag{1}$$

证明 (1) 的证明基于这样的简单事实: G 的顶点的 k 着色可以分成两种独立的情况: (a) 给 x 和 y 分配不同的颜色; (b) 给 x 和 y 分配相同的颜色. 对于情况 (a), G 的 k 着色数等于 $G_{+(x,y)}$ 的 k 着色数, 因为加入边 (x, y) 迫使 x 和 y 有不同的颜色. 同样, 对于情况 (b), G 的 k 着色数等于 $G_{x=y}$ 的 k 着色数. ◇

推论 设 x, y 是图 G 的两个相邻顶点. 设 $G_{-(x,y)}$ 是通过从 G 中删除边 (x, y) 而得到的图. 设 $G_{x=y}$ 是如定理 6 那样, 通过把 G 中两个顶点 x 和 y 粘合成一个顶点而得到的图. 这时有

$$P_k(G) = P_k(G_{-(x,y)}) - P_k(G_{x=y}) \tag{2}$$

证明 设 H 是图 $G_{-(x,y)}$. 此时顶点 x, y 在 H 中不相邻. 观察发现, $H_{+(x,y)} = G$. 同样, $H_{x=y} = G_{x=y}$. 对图 H 和不相邻的顶点 x, y 运用等式 (1) 得到

$$P_k(H) = P_k(H_{+(x,y)}) + P_k(H_{x=y}) \tag{3}$$

用 $G, G_{-(x,y)}, G_{x=y}$ 重新表示等式 (3), 我们得到

$$P_k(G_{-(x,y)}) = P_k(G) + P_k(G_{x=y}) \tag{4}$$

求解 (4) 中的 $P_k(G)$, 得到等式 (2). ◇

现在, 利用定理 6 重做例 1 中的问题 (c).

运用定理 6 重做例 1(c)

4 回路 $C_4 = a - b - c - d - a$ 的色多项式是什么?

首先使用等式 (1), 然后使用等式 (2) 来确定 C_4 的色多项式. 我们把等式 (1) 运用到带有一对不相邻顶点 a, c 的 C_4. 图 $(C_4)_{+(a,c)}$ 是由两个三角形 (a, b, c) 和 (a, c, d) 组成的, 它们有一条公共边 (a, c). 三角形 (a, b, c) 是一个完全图, 根据定理 5(b), 可以用 $k(k-1)(k-2)$ 种方法 k 着色完全图三角形 (a, b, c). 接下来对于 d 则可以选择 $k-2$ 种着色颜色 (避开 a 和 c 的颜色). 所以 $P_k((C_4)_{+(a,c)}) = k(k-1)(k-2)^2$. 图 $(C_4)_{a=c}$ 是 3 顶点路径 $b - ac - d$ (其中 ac 是把 a 和 c 粘合而成的顶点). 所以根据定理 5, $P_k((C_4)_{a=c}) = k(k-1)^2$. 综合这两个色多项式, 我们有 $P_k(C_4) = k(k-1)[(k-1) + (k-2)^2]$.

接下来, 我们把等式 (2) 运用到有两个相邻顶点 a 和 b 的 C_4 上. 根据定理 5(a), 图 $(C_4)_{-(a,c)}$ 是 4 顶点路径 $b-c-d-a$ 且色多项式是 $k(k-1)^3$. 图 $(C_4)_{a=b}$ 是三角形 (ab,c,d), 且色多项式是 $k(k-1)(k-2)$. 于是根据等式 (2),

$$\begin{aligned}P_k(C_4) &= P_k((C_4)_{-(a,b)}) - P_k((C_4)_{a=b}) \\ &= k(k-1)^3 - k(k-1)(k-2) = k(k-1)[(k-1)^2-(k-2)]\end{aligned} \qquad \square$$

注意, 使用等式 (1) 得到的答案形式与我们原来使用特殊方法得到问题 (c) 的答案形式完全一样. 使用等式 (2) 得到的答案却稍有不同, 但却是代数等价的.

练 习 2.4

练习概况 大部分练习都是着色理论结果的证明.

1. 利用每个可平面图都有一个度最多为 5 的顶点这一事实, 给出每个可平面图都可以 6 着色的简单归纳证明. 按定理 5 开头的说明进行.

2. 证明: 有 8 个顶点和 13 条边的可平面图 G 不能 2 着色. (提示: 使用 1.4 节的结果证明 G 一定 包含三角形.)

3. 假设 G 有 k 个连通分量, 标记为 $G_1, G_2, \cdots, k$. 设 $\chi(G_i)$ 表示第 i 个连通分量的色数. 用各 $\chi(G_i)$ 表示色数 $\chi(G)$ 的公式.

4. 设 G 是每个顶点的度至少为 3 的图. 对下面的论断给出证明或反例: 至少需要 3 种颜色着色 G 的顶点.

5. 什么样的图的最小顶点着色和最小边着色的大小都等于 2?

6. 利用归纳法证明: 多边形的任意三角剖分将至少有两个度为 2 的顶点. (提示: 在某条弦 e 处把这个三角剖分图划分成两个三角剖分图.)

7. **(a)** 利用 2.3 节的推理模式, 解释如何 3 着色多边形的任意三角剖分.

(b) 使用问题 (a) 的讨论证明: 不计颜色的名称变化, 多边形的三角剖分的 3 着色是唯一的.

8. **(a)** 如果 q 是图 G 中最大独立集合的大小, 证明 $\chi(G)q \geqslant n$, 其中 n 是 G 的顶点数.

(b) 如果在有 n 个顶点的图 G 中, 顶点的最小度是 d, 那么利用 (a) 的结果证明 $\chi(G)(n-d) \geqslant n$, 因此 $\chi(G) \geqslant n/(n-d)$.

9. 一个图是色临界的, 如果删除任意顶点都将减少这个图的色数. 证明: 每个 k 颜色色临界图 G 都有下面的性质:

(a) G 是连通的.

(b) G 的每个顶点的度 $\geqslant k-1$.

(c) G 没有其删除将使 G 不再连通的顶点.

10. 证明: G 可以对其边进行 k 着色, 当且仅当 G 的线图 (参见 2.1 节练习 16) $L(G)$ 可以对其顶点进行 k 着色.

11. 设 $\bar{G}$ 是 G 的补, 证明:

(a) $\chi(G)+\chi(\bar{G}) \leqslant n+1$ (提示: 使用归纳法);

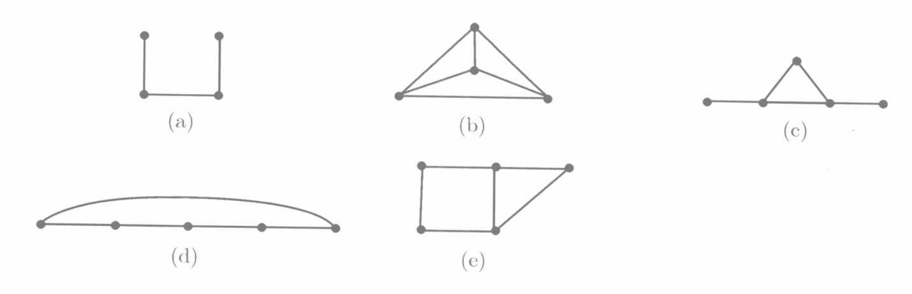
(b) $\chi(G)\chi(\bar{G}) \geqslant n$;

(c) $\chi(G)+\chi(\bar{G})\geqslant 2\sqrt{n}$.

12. 证明：如果可平面图的每一个区域都有偶数条边界边, 那么它的顶点可以 2 着色.

13. 证明：任意可平面图 G 的色多项式都不可能形如 $P_k(G)=(k^2-6k+8)Q(k)$, 其中对于 $k>0, Q(k)$ 是正整数.

14. 确定下面各图的色多项式 $P_k(G)$:



(a) (b) (c) (d) (e)

15. 证明：一个图可以 k 着色, 当且仅当可以给其边确定方向, 使得它没有有向回路且其最长路径的长度是 $k-1$.

16. 利用每一个少于 12 个顶点的可平面图都有一个度 $\leqslant 4$ 的顶点 (1.4 节的练习 19) 的事实, 证明每一个少于 12 个顶点的可平面图可以 4 着色.

17. 证明：最多有两条奇长度回路的图可以 3 着色.

18. 证明：对于任意的正整数 k, 存在 $\chi(G)=k$ 且不含三角形的图 G.

(a) 对色数 k 用归纳法来进行证明. 首先, 对于 $k=3$, 我们使用 5 回路图 G_3 给出存在性证明.

(b) 假设能够构造 $\chi(G_k)=k$ 且不含三角形的图 G_k, 我们这样构造 G_{k+1}：首先做 k 份 G_k 的拷贝, 然后加入 $(n_k)^k$ 个顶点, 其中 n_k 是 G_k 的顶点数. 每一个新顶点的邻居集合是由 G_k 的每一个拷贝中的一个顶点构成的不同 k 元组组成的. 确认这个新图就是我们需要的 G_{k+1}.

2.5 小结及参考文献

本章给出了两个重要的图论概念：覆盖循环或回路和着色. 2.1 节讨论了欧拉圈, 即, 正好遍历每一个边一次的圈. 2.2 节讨论了哈密顿回路, 即, 正好访问每一个顶点一次的回路. 这两种类型的覆盖边集合都自然地出现在运筹学的行程问题中. 尽管欧拉圈和哈密顿回路在定义上类似, 但是, 作为图论问题, 确定图中欧拉圈和哈密顿回路是否存在的方法却完全不同. 欧拉圈定理使我们可以迅速确定欧拉圈是否存在. 而另一方面, 除特殊情况外, 为了确定哈密顿回路存在或不存在, 我们只能通过大量的系统研究来穷尽构建哈密顿回路的所有方法.

2.3 节介绍了使用特定着色方案的图着色以及着色的一些应用. 2.4 节展示了着色理论的样本, 重点是任意可平面图可以是 5 着色的证明. 1976 年 Appel 和 Haken 所证明的更强的定理 (见文献 [1]), 任何可平面图都可以 4 着色, 激发了 100 年来

图论的诸多研究. 对四色定理证明的探索, 导致了用哈密顿回路和其性质得到验证的其他图概念来重新描述这一定理.

1736 年欧拉对欧拉回路的分析是图论的第一篇论文. 欧拉的论文 (参见 [3]) 读起来非常有趣. 看看在当时缺乏现代图论术语的情况下的欧拉的笨拙写作是非常有启发意义的. 哈密顿回路的概念的首次使用出现在 1771 年由 A.Vandermonde 写的一篇论文中, 在这篇论文中, 他给出了一个棋子马的移动序列, 通过这个序列, 马可以不重复地遍游棋盘的所有位置. 名字 “哈密顿” 指的是 W.Hamilton, 他在代数方面的研究促使他考虑十二面体边上的特殊回路和路径 (2.2 节中练习 2(c) 中的图). 哈密顿甚至拥有一个在市场上销售的游戏, 其中就涉及寻找十二面体上的哈密顿回路 ([3] 重印了哈密顿对这一游戏给出的说明). 关于四色问题的历史, 它的重述以及 Appel 和 Haken 给出的最终解决方案, 参见 Barnette[2].

1. K. Appel and W. Haken, “Every planar map is 4-colorable, ” *Bull. Am. Math. Soc.* 82(1976), 711–712.

2. D. Barnette, *Map Coloring and The Four Color Problem*, Mathematical Association of America, Washington, DC, 1984.

3. N. Biggs, E. Lloyd, and R. Wilson, *Graph Theory 1736–1936*, Cambridge University, Cambridge, 1999.

4. J. Bondy and U. Murty, *Graph Theory with Applications*, American Elsevier, New York, 1976.

5. J. O’Rourke, *Art Gallery Theorems and Algorithms*, Oxford University Press, New York, 1987

补充材料：四方柱问题的图模型

这一补充材料给出由 Blanche Decartes (据说是著名的图论学家 W.Tutte 的笔名) 设计的四方柱问题的一个巧妙的解决方案. 这个解决方案使用哈密顿回路的一个扩展形式, 即所有顶点都被一组没有相连顶点的回路覆盖, 而不是被单一回路覆盖.

四方柱问题包括 4 个立方体, 这些立方体的各面被着色成 4 种颜色中的一种：红 (R), 白 (W), 蓝 (B), 绿 (G). 第 i 个立方体的 6 个面标记为：f_i(前面), l_i (左面), b_i (后面), r_i (右面), t_i (上面), u_i (下面). 在图 2.20 中, 第一个立方体如下着色：$l_1 = B, r_1 = W, f_1 = R, b_1 = R, t_1 = G, u_1 = B$. 这一问题的目标是把 4 个立方体堆成一个柱子 (立方体 1 放在立方体 2 的上面, 立方体 2 在立方体 3 的上面,$\cdots\cdots$, 以此类推), 使得对于每一种颜色, 这个柱子的每一个侧面都有一个面是该颜色的. 例如, 图 2.21 给出了 2.20 所给立方体的一个四方柱解.

我们使用图 2.20 中的立方体进行讨论. 确定如何把四个立方体排成四方柱是非常困难的任务. 观察发现, 一个立方体有 24 种对称, 因此总共可以构建 24^4=331 776 种不同的柱子. 尽管面颜色的对称性及每一侧面都没有重复颜色的限制将排除很多可能性, 但枚举树搜索仍然非常之大. 很容易编写出这一搜索的计算程序. 幸运的是, 可以按下面的方式用一个有 4 个顶点的图来对这一问题建模, 使得可以通过几分钟的观察给出这一问题的图论重述.

在给出这个图模型之前, 需要讨论这个问题的一个简单分解原则. 在一个柱子中排列各立方体, 使得这个柱子的左右侧面对每种颜色有一个面的工作与使它的前后面对每一种颜色有一个面的工作 “无关”. 这里, “无关” 的意思是一旦安置好立方体 i, 使得给定的一对对面在这个柱子

的左面和右面上, 那么立方体 i 上的其余对立面对可以在这个柱子的前面和后面上.

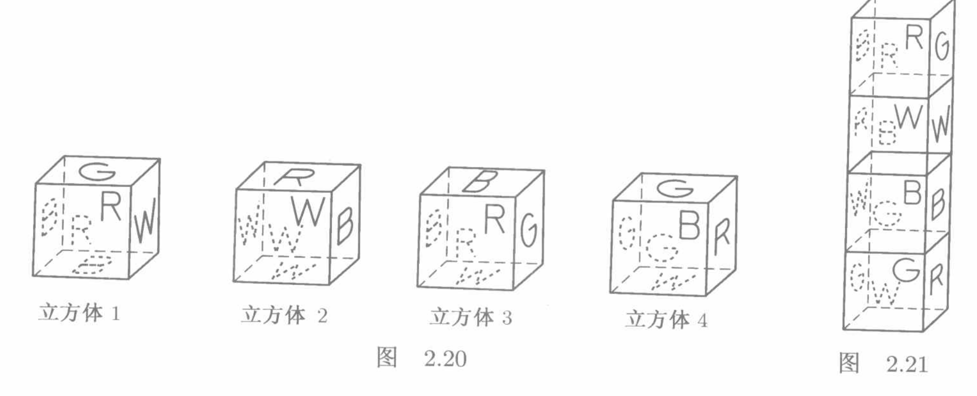


图 2.20

图 2.21

通过旋转立方体 i 来得到四方柱的解时, 设 l_i^*, r_i^*, f_i^*, b_i^* 是立方体 i 在这个柱子的 4 个侧面上可视的 4 个面的颜色. 假设 $l_1^* = t_1 = G$, $r_1^* = u_1 = B$, 即, 图 2.20 中的立方体 1 的上面 t_1 是绿色, 它被重新旋转到左侧 l_1^*, 而下面 u_1 的蓝色被重新旋转到右侧 r_1^*. 于是, 通过关于这两个侧面的中心旋转立方 1, 我们可以得到 $f_1^* = B$, $b_1^* = W$ 或者 $f_1^* = R$, $b_1^* = R$ 或者 $f_1^* = W$, $b_1^* = B$, 这是前面和后面所有剩余的选择. 因为对于其他 3 个立方体, 这种左–右和前–后的 "独立性" 是同样成立的, 所以我们看到这个问题可以分解成两个不相关的问题.

分解原则

(1) 为四方柱的右侧和左侧挑选出每一个立方体的一对对立面, 使得这个柱子的这两个侧面对每一种颜色有一个面;

(2) 为四方块柱的前后侧面挑选出每一个立方体的不同对对立面, 使得这两个侧面对每一种颜色有一个面.

现在, 我们已准备好给出图模型 (由 F.de Carteblanche 提出). 实际上, 我们使用多重图 (其中允许有多重边和循环). 4 种颜色的每一种作为一个顶点. 对立方体 i 上的每一对对立面之间有一条标签为 i 的边, 它连接表示这两个对立面的颜色的两个顶点. 对于立方体 1 上的对立面 $l_1 = B, r_1 = W$, 我们在顶点 B 和顶点 W 之间画出一条标签为 1 的边; 对于 $f_1 = R$, $b_1 = R$, 我们在顶点 R 处画出一条标签为 1 的循环边; 对于 $t_1 = G$, $u_1 = B$, 我们在顶点 G 和 B 之间画出一条标签为 1 的边. 同样, 对于其他立方体我们画出相应的边. 图 2.22 给出了对应于图 2.20 中的立方体的多重图.

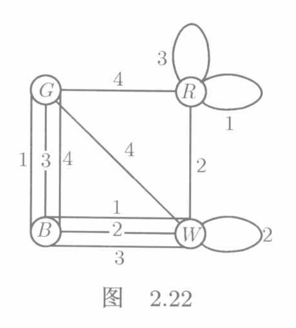


图 2.22

根据分解原则, 可以把这个问题分解成左–右部分和前–后部分. 首先, 我们只考虑左–右部分, 即对每一个立方体寻找一对对立面, 使得四方柱的左侧面和右侧面对每一种颜色有一个面.

对这个左–右问题再做一些简化: 我们在每个立方体中寻找一个对立面对, 构成一个 4 个对立面对的集合, 使得在这个总共有 8 个面的集合中, 每一种颜色出现两次. 后面将展示如何确保每一种颜色在左侧面出现一次, 在右侧面出现一次. 因为一种颜色对应于一个顶点, 一个立方体

对应于一个边号码 (标签), 一对对立面对应于一条边, 可以把这一简化的左–右问题重述成下面的图论问题：寻找 4 条边, 对每个数字有一条边, 使得这 4 条边的 8 个端顶点的集合包含每一个顶点两次.

关于 4 条边端顶点的条件等价于要求在这 4 条边形成的子图中每一个顶点的度为 2(自循环在其顶点处的度计数为 2). 注意, 所有顶点的度都为 2 的子图是回路或不相交回路的集合 (自循环是长度为 1 的回路). 在有 n 个顶点的多重图中, 形成不相交简单回路的 n 条边的集合称为**因子** (factor). 观察发现, 因子是哈密顿回路的自然扩展. 在四方柱模型中, 我们使用术语**标签因子** (labeled factor) 来表示每一个边号码出现一次的因子. 在图 2.23 中, 我们对图 2.22 中的多重图给出了 3 个可能的标签因子.

四方柱问题的图模型的简化左–右问题化简为：寻找一个标签因子.

下面说明如何把标签因子转化成这样的立方体排列, 其中, 柱子的左右侧面对每一种颜色有一个面. 我们给每一条回路中的各边指定一个一致的方向, 例如, 按顺时针方向 (参见图 2.23). 考虑图 2.23a 中的标签因子. 在沿着这条回路环绕时, 我们这样排列每一个立方体：以一条边的末端顶点的颜色作为这个立方体左侧的颜色, 以头部顶点的颜色作为右侧的颜色①. 可以从回路的任意边开始, 我们选择标签为 1 的边. 在图 2.23a 中, 标签为 1 的边从 B 到 G, 所以我们排列立方体 1, 使得 l_1^*(在堆中立方体 1 的左侧面)$=B$, $r_1^*=G$ (参见图 2.24). 沿着这条回路中标签为 1 的边 (在回路的顺时针方向), 接下来我们遇到从 G 到 R 标签为 4 的边. 由此, 我们排列立方体 4, 使得 $l_4^*=G$, $r_4^*=R$. 接着来, 到达从 R 到 W 标签为 2 的边, 其后是从 W 到 B 标签为 3 的边. 所以, 我们排列立方体 2 和立方体 3, 使得 $l_2^*=R$, $r_2^*=W$, $l_3^*=W, r_3^*=B$. 图 2.24 给出了按上面的描述排列成的立方体柱的左右侧面.

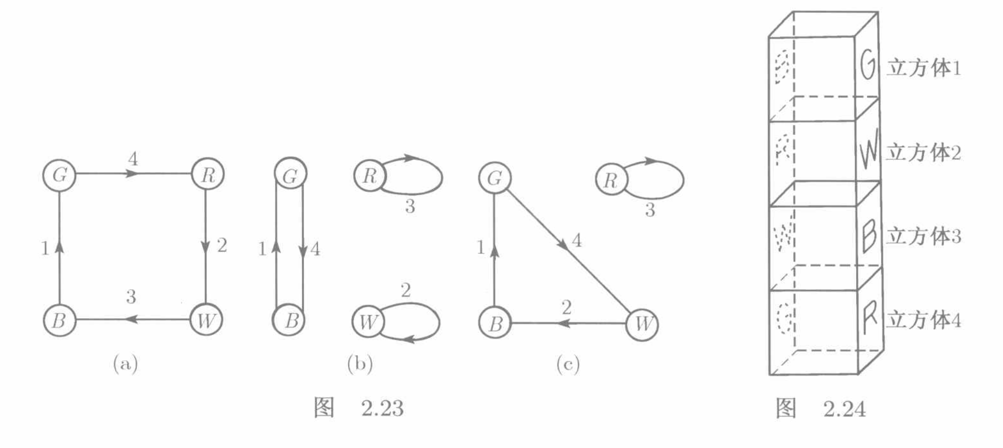


图　2.23

图　2.24

这个过程保证每一种颜色在一个侧面只出现一次, 因为每一个顶点 (颜色) 处于一条边的头部和一条边的尾部. 如果已经选定了图 2.23b 中的一个标签因子, 那么对所有 3 条回路, 我们应该使用顺时针游历过程, 对一条回路产生 $l_1^*=B, r_1^*=G, l_4^*=G, r_4^*=B$, 而对于两个自循环产生 $l_3^*=r_3^*=R, l_2^*=r_2^*=W$.

现在, 我们已准备好解决四方柱问题. 使用标签因子重述分解原则：

① 有向边 $(\overrightarrow{a,b})$ 的头部顶点是 a, 尾部顶点是 b. —— 译者注

四方柱问题的图论形式

(1) 在四方柱问题的图中, 分别为左–右侧面和前–后侧面寻找一个边不相交的标签因子;

(2) 使用顺时针游历过程确定每一个立方体的左–右侧面和前–后侧面的排列.

通过观察来寻找两个不相交标签因子并不是很困难. 图 2.23 中的 3 个标签因子都使用相同的标签为 1 的边 (即 B 和 G 之间的边). 所以这 3 个标签因子中的任意两个因子都相交. 最简单的方法是寻找一个标签因子, 删除它的边, 然后再寻找第二个标签因子. 如果没有找到, 那么就从另一个不同的标签因子开始. 例如, 假设我们使用图 2.23a 中的标签因子作为第一个因子. 从图 2.22 中删除它的边之后, 我们很容易找到图 2.25 所示的第二个标签因子 (读者应该能够找到第二个因子).

使用图 2.23a 中的因子来排列左右侧面, 如图 2.24 所示. 接下来关于每一个立方体的左右侧面的中心旋转每一个立方体, 并按照图 2.25 中回路的顺时针遍历来排列前后面. 从标签为 2 的边开始, 我们标记 $f_2^* = W, b_2^* = B, f_3^* = B$, $b_3^* = G, f_4^* = G, b_4^* = W$, $f_1^* = b_1^* = R$. 图 2.26 给出了这个四方柱问题的一个解. 现在就去购买或借来一套四方柱, 向你的朋友展示一下从这本书学来的一些切实有用的知识吧.

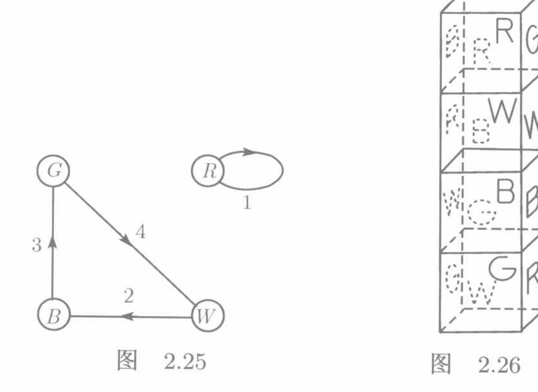
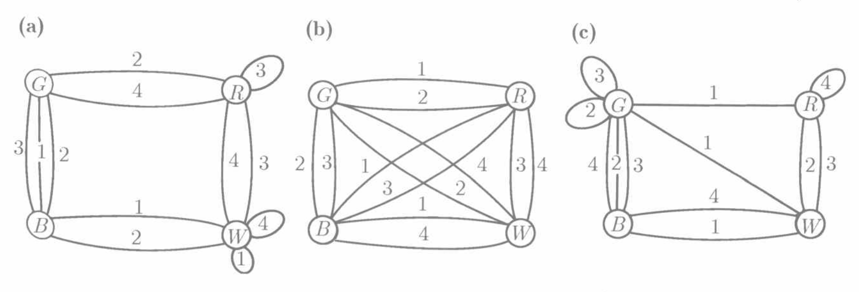


图 2.25　　图 2.26

补 充 练 习

1. 对于图 2.22 中的多重图, 寻找图 2.23 和图 2.25 的标签因子之外的其他标签因子. 对这一过程给出讨论来说明不可能存在其他标签因子.

2. 对于与图 (a) 有关的四方柱游戏, 寻找所有的四方柱解 (即不相交标签因子对).



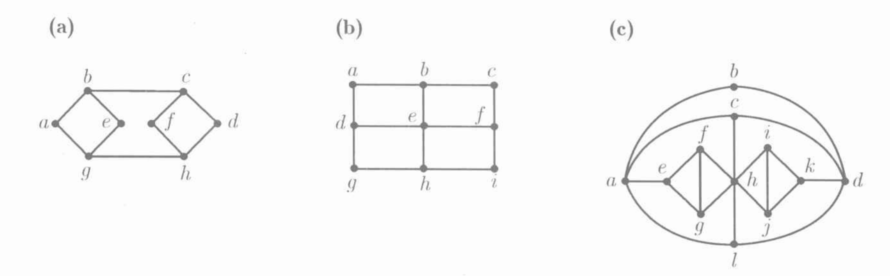
3. 对于与图 (b) 有关的四方柱游戏, 寻找所有的四方柱解.

4. 证明: 上面所示的四方柱图 (c) 没有解, 即, 不存在不相交的标签因子对.

5. **(a)** 如果图 G 有哈密顿回路, 那么它也一定有因子吗? 证明其为真或给出反例.

(b) 对于有欧拉圈的图, 重做问题 (a).

6. 对于下面各图, 如果存在的话, 寻找一个因子:



7. 证明: 因子的定义表明, 图中的每个顶点都正好与因子的两条边 (或一个自循环) 相连.



第3章　树和搜索

3.1　树的性质

使用最为广泛的特殊图就是**树**(tree). 树的定义方式有两种. 在无向图中, 树是没有回路的连通图. 另外, 也可以把树定义为这样的图, 它有一个称为**根**(root) 的指定顶点, 使得从这个根到树的任意其他顶点有唯一一条路径. 可以像无向图那样把第二种定义运用于有向图. 定理 1 证明这两种定义的等价性.

直观上, 树看起来就像自然界中的树一样. 参见图 3.1 中的树的例子. 被标记为 a 的顶点是图 3.1 的每一棵树的根. 练习 5(以另一种方式) 证明上面两种定义的等价性. 为了说明等价性, 请看图 3.1a 中的树, 把边 $(g^{\rightarrow}, h)$ 加入这棵树将创建一条回路 (忽视边的方向), 同时将创建第二条从根 a 经由 g 到 h 的路径.

树是组织信息和搜索过程的极为有力的工具. 本章概述可以用树来表示并分析搜索过程的若干不同应用背景. 这些应用背景包括迷题求解 (3.2 节)、"旅行商"问题求解 (3.3 节) 以及排序列表 (3.4 节). 在第 4 章, 树在几个网络算法中扮演着重要的角色. 本节给出树的一些基本性质, 然后介绍使用树时的一些术语. 我们证明与树的计数有关的有用公式, 并说明这些公式的一些用途.

观察发现, 如果树是无向图 (没有有向边), 那么任意顶点都可以是根. 例如, 图 3.1b 中的树画成了看似以 a 为根的树, 但是把它画成图 3.1c 的样子时, 这棵树就没有自然可以看成根的顶点, 所以, 任意顶点都可以是根.

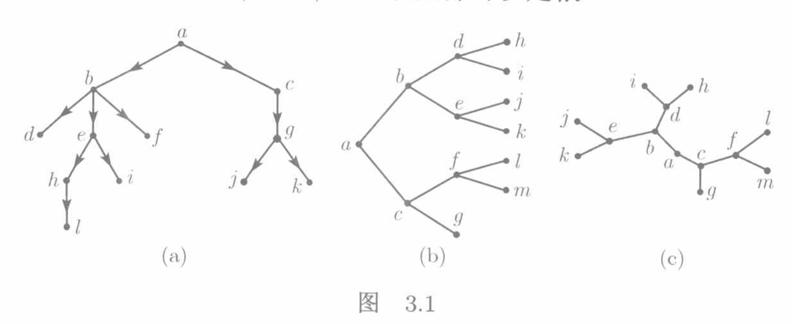


图　3.1

在本章的大部分内容中, 我们都将使用有有向边的树. 根据通常的术语, 我们称有向树为**根树**(rooted tree). 根树 T 有唯一的根, 因为如果顶点 a 和 b 都是 T 的根, 那么存在从 a 到 b 和从 b 到 a 的路径, 它们形成一个回路. 无向树在没有特殊根的意义下是无根的. 通过选出无向图的一个顶点当作它的根, 然后对所有边赋予

以方向为离开这个根, 就可以把这个无向树变成有向树. 例如, 为了使图 3.1b 中的无向树以顶点 a 为根, 我们简单地给所有边赋予从左到右的方向.

画根树 T 的标准方法是把根 a 放置在图的顶部. 然后, 把与 a 相邻的顶点放置在 a 的下一层上, 以此类推, 如图 3.1a 所示. 我们说根 a 位于层次 0, 这棵树上的顶点 b 和 c 位于层次 1, 顶点 d, e, f, g 位于层次 2, 以此类推. 顶点 x 在 T 的**层次数**(level number) 是从根 a 到 x 的 (唯一) 路径的长度.

对于根树 T 中根之外的任意顶点 x, x 的**双亲**(parent)是有进入到 x 的边 $(y\neg, x)$(指向 x 的唯一边) 的顶点 y. x 的**孩子**(children) 是有从 x 指向 z 的边的顶点 z. x 的孩子的层次数比 x 的层次数大 1. 有相同双亲的两个顶点是**兄弟**(sibling). 双亲 — 孩子关系可以扩展到顶点的祖先和后代. 在图 3.1a 中, 顶点 e 的双亲是 b, h 和 i 则是它的孩子, d 和 f 为它的兄弟, a 则是它的另一个祖先, l 是它的另一个后代. 观察发现, 树 T 中每一个顶点 x 是 x 和它的后代构成的子树的根. 为了方便参考, 本书末尾给出了一个与树有关的术语表.

现在, 我们证明在本章开始给出的 (无向) 树的两个定义, 以及树的另两个其他特征描述, 是等价的.

定理 1 *设 T 是连通图. 这时, 下面的陈述等价:*

(a) *T 没有回路.*

(b)*设 a 是 T 中的任意顶点. 那么对于 T 的其他任意顶点 x, a 与 x 之间存在唯一一条路径 P_x.*

(c) *T 的任意一对不同的顶点 x, y 之间存在唯一一条路径.*

(d)*在删除 T 的任意一条边都使 T 不再连通的意义下, T 是最小连通的.*

证明 首先, 显然 (b) 和 (c) 是等价的: 可以设 (b) 中的顶点 a 是 (c) 中的顶点 y. 我们证明 (a) 可以推导出 (d), (d) 可以推导出 (c), (c) 可以推导出 (a).

(a)→(d): 假设删除某条边 (a, b) 后 T 仍连通. 那么图 $T-(a, b)$ 包含 a 与 b 之间的一条路径 P. 然而, P 的边加上边 (a, b) 形成一个回路, 这与 (a) 矛盾.

(d)→(c): 假设 T 的两个顶点 a 与 b 之间有两条不同的路径 P_1 和 P_2. 令 $e=(u, v)$ 是从 a 开始的 P_1 上的第一条不在 P_2 上的边 (有可能 $u=a$ 或 $v=b$). 根据 (d), $T-e$ 是非连通的, 所以 u 和 v 必须在两个不同的连通分量中. 但是, 沿着 P_2 从 u 到达 b, 然后再返回来沿着 P_1 从 b 到达 v, 这将创建连通 u 和 v 的路径 Q. 注意: 为了确保 Q 没有重复任何边, 我们实际上应该沿 P_2 行走, 直到遇到也在 P_1 上的第一个顶点 w(这样的 w 有可能在达到 b 之前出现), 然后再沿着 P_1 到达 v.

(c)→(a): T 不能包含回路, 因为回路的边为这条路径上的任意两个顶点提供了连接它们的两条不同路径. ◇

作为最小连通图, 树的边尽可能少. 下面的定理给出 n 顶点树的边数的简单公式.

定理 2 *有 n 个顶点的树有 $n-1$ 条边.*

证明 假设这棵树是根树; 如果它是无向的, 我们可以像上面描述的那样使它成为根树. 可以把顶点 x 与从它的双亲 y 进来的边 $(y^{\rightarrow}, x)$ 配对. 因为根以外的每一个顶点有唯一一条进入边, 而且存在 $n-1$ 个非根顶点, 因此有 $n-1$ 条边. ◇

T 的没有孩子的顶点称为 T 的**叶子**(leave). 有孩子的顶点称为 T 的**内部**(internal) 顶点. 如果根树的每一个内部顶点都有 m 个孩子, 则称 T 是 **m 元树**(m-ary tree). 如果 $m=2$, 则 T 是**二叉树**(binary tree).

定理 3 *设 T 是有 n 个顶点且有 i 个内部顶点的 m 元树. 那么 $n=mi+1$.*

证明 树上根之外的每一个顶点是唯一一个顶点 (它的双亲) 的孩子. i 个内部顶点的每一个都有 m 个孩子, 所以总共有 mi 个孩子. 加上一个不是任意顶点的孩子的根顶点, 我们得到 $n=mi+1$. ◇

推论 *设 T 是有 n 个顶点的 m 元树, 它包含 i 个内部顶点和 l 个叶子. 如果知道 n, i 或 l 中的一个, 那么根据下面的公式可以给出其他两个参数:*

(a) *已知 i, 则 $l=(m-1)i+1$ 且 $n=mi+1$.*

(b) *已知 l, 则 $i=(l-1)/(m-1)$ 且 $n=(ml-1)/(m-1)$.*

(c) *已知 n, $i=(n-1)/m$ 且 $l=[(m-1)n+1]/m$.*

推论直接得之于公式 $n=mi+1$(定理 3) 和 $l+i=n$ 的事实. 其详细证明留作练习 8.

例 1 如果有 56 个人报名参加网球循环赛, 那么在这次循环中要举行多少场比赛?

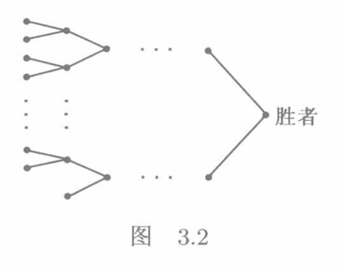


图 3.2

循环赛是以类似于二叉树的形式进行的. 选手是叶子, 比赛是内部顶点. 参见图 3.2. 给定 $l=52$ 和 $m=2$, 我们根据推论中的部分 (b) 确定 $i: i=(l-1)/(m-1)=(56-1)/(2-1)=55$ 场比赛. □

例 2 假设为传达学校放假的消息, 100 名家长之间要进行电话联络. 联络由一位家长发起, 他给另 3 位家长打电话. 这 3 位家长再各自给其他 3 位家长打电话, 以此类推. 有多少位家长需要打电话? 如果有 200 位家长, 又有多少家长需要打电话?

这样的电话联络是一棵有 100 个顶点的根树. 边对应通话, 内部顶点对应于打电话的家长. 因为我们知道 $n=100$, 而且知道这棵树是 3 重的 (3 元树), 可以利用前面的推论 (c) 确定打电话的人数 $i: i=(n-1)/m=(100-1)/3=33$.

当对有 200 个人的团体重复这一计算时, 我们得到 $i=(200-1)/3=66\frac{1}{3}$ 个内部顶点. 根据定理 3, 3 元树的顶点数 n 一定等于 $3i+1$, 其中 i 是一个整数, 即, $n\equiv 1(\mathrm{mod}\ 3)$. 但是 $200\equiv 2(\mathrm{mod}\ 3)$, 所以我们不可能有一棵真正的 3 元树. 199 位或者 202 位家长将会给出一棵 3 元树. 作为一个实际问题, 对于 200 位家长, 将有 66 位家长每人打 3 通电话, 而有 1 位家长只打 1 通电话 (有一个内部顶点只有一个孩子). □

根树的**高度**(height) 是从根开始的最长路径的长度, 或等价地, 顶点的最大层次数. 如果高度为 h 的根树的所有叶子都位于层次 h 或层次 $h-1$, 那么称它为**平衡的**(balance). 平衡树是 "好" 树. 例 2 中的电话联络树应该是平衡的, 以便把信息尽快地传达给每一个人. 网球循环赛树应该是平衡的, 以求公平; 否则就会有一些选手在比赛场次少于另一些选手的情况下进入决赛. 做 m 元平衡树将使其高度最小 (参见练习 12). 图 3.1b 中以 a 为根的树是高度为 3 的平衡二叉树.

定理 4 设 T 是高度为 h 且有 l 个叶子的 m 元树. 那么:

(a) $l \leqslant m^h$, 且如果所有叶子的高度都是 h, 那么 $l = m^h$.

(b) $h \geqslant \lceil \log_m l \rceil$, 而且如果这棵树是平衡的, 那么 $h = \lceil \log_m l \rceil$.

证明 符号 $\lceil r \rceil$ 表示大于等于 r 的最小整数, 即 $\lceil r \rceil$ 把 r 入到下一个整数.

(a)高度为 1 的 m 元树有 $m^1 = m$ 个叶子 (根的孩子). 现在, 我们对 h 使用归纳法证明高度为 h 的 m 元树至多有 m^h 个叶子, 如果所有叶子都处于层次 h 上, 则 $l = m^h$. 高度为 h 的 m 元树可以分解成以根的 m 个孩子为根的 m 棵子树. 参见图 3.3. 这 m 棵子树的高度至多是 $h-1$. 根据归纳法, 每棵子树的高度至多是 m^{h-1}. 于是 m 棵子树至多有 $m \times m^{h-1} = m^h$ 个叶子. 如果所有叶子都处在高度 h 上, 则正好有 m^h 个叶子.

(b)在不等式 $l \leqslant m^h$ 的两边, 以 m 为底取对数, 得到 $\log_m l < h$. 因为 h 是整数, 我们有 $\lceil \log_m l \rceil \leqslant h$. 如果这棵树是高度为 h 的平衡树, 那么 l 的最大可能值是 $l = m^h$(如果所有叶子都在层次 h 上), 而且最小可能值是 $l = (m^{h-1} - 1) + m$(有 m 个叶子在层次 h 上, 而其余叶子在层次 $h-1$ 上). 所以 $m^{h-1} < l \leqslant m^h$. 对这个不等式的两边以 m 为底取对数, 得 $h - 1 < \log_m l \leqslant h$ 或者 $h = \lceil \log_m l \rceil$. ◇

树的最常见的应用是搜索. 下面两个顺序搜索的例子说明了树在搜索中的应用, 其中一个是基本的计算机科学问题, 另一个是逻辑问题.

例 3 重新分析 1.1 节例 2 所讨论的词典查找问题. 通过把一个未知词 (数)X 与这个词所属的集合 (词典) 中的诸多单词进行比较来识别它. 现在, 我们的比较检测有 3 个分支方向 (小于, 等于, 大于). 这一检测过程可以用二叉树或几乎是二叉树的树来表示. 如果 X 是字母表中前 14 个字母中的一个, 那么图 3.4 就是这样的一棵二叉搜索树. 检测过程的各个阶段要被测试的字母是这棵树的顶点. 图 3.4 所示的这个检测过程开始于把 X 与 H 相比较. 当 X 小于这个字母时, 就取这个顶点左侧的边; 当 X 比 H 大时, 就取这个顶点右侧的边. 如果有偶数个顶点, 那么这样的树会有一个只有一个孩子的顶点 (如图 3.4 中顶点 N 的情况). 我们把检测列表的中间字母作为根 (对于本例, 根是 G 或者 H). 根的左孩子是左子树的中间字母 (此例是 D), 以此类推.

为了使识别任意 X 所需的检测次数最少, 即搜索树的高度最低, 我们应该使这棵树平衡. 假设已知 X 属于一个有 n 个 "单词" 的集合. 识别 X 所需的最多检

测次数是多少呢？根据定理 3 的推论，有 n 个顶点的二叉搜索树有

$$l=\frac{(2-1)n+1}{2}=\frac{n+1}{2}$$

个叶子. 因此, 识别 X 所需的最多检测次数是有 $1/2\ (n+1)$ 个叶子的平衡搜索树的高度. 根据定理 4, $h=\lceil\log_2(n+1)/2\rceil=\lceil\log_2(n+1)\rceil-1$. □

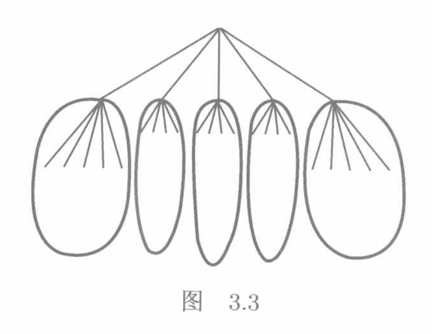


图 3.3

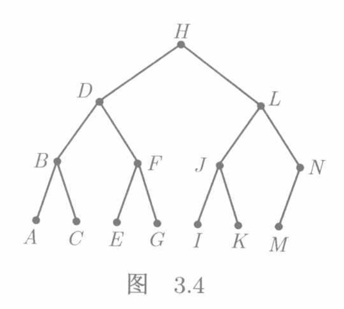


图 3.4

例 4 一个著名的逻辑问题说的是, n 枚硬币中有一枚假币, 它或者太重或者太轻, 还有一台用来比较任意两组硬币重量的天平 (这台天平或者倾向右边, 或者倾向左边, 或者保持平衡). 对于给定值 n, 我们寻找使用最少称重次数发现假币的过程. 有时候, 我们已经知道这枚假币是太重还是太轻. 如果我们知道这枚假币太轻, 那么对于 n 枚硬币需要多少次称重呢？

我们的检测过程将形成一棵树, 其中根是第一次检测, 其他内部顶点是其余检测, 叶子是答案, 即, 哪一枚硬币是假币. 参见图 3.5 中 8 枚硬币的检测过程. 我们从 1 到 8 对硬币编号. 在一次检测中, 如果天平左边较轻, 就取左侧的边; 如果天平左右平衡, 就取中间的边; 如果天平右边较轻, 就取右侧的边. 注意, 当称重 4 和 5 时 (已知 1, 2, 3, 6, 7, 8 不是轻硬币), 天平不能平衡.

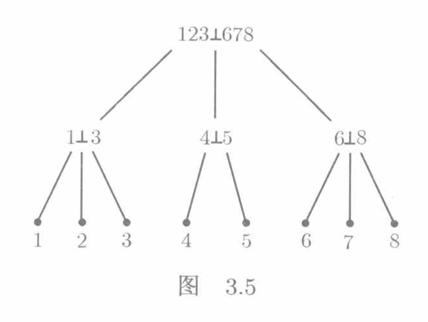


图 3.5

这个检测树 (几乎) 是三元树, 因为有 n 枚硬币, 因此有 n 个叶子, 即, 假币有 n 种不同的可能性. 定理 4 告诉我们, 包含 n 个叶子的检测树的高度至少是 $\lceil\log_3 n\rceil$. 对于轻假币问题, 我们可以通过持续地把已知包含假币的子集分解成 3 个几乎相等的堆, 并比较等大小的堆, 来达到上面的树高边界, 如图 3.5 所示.

如果假硬币可能太轻也可能太重, 那么这个问题就变得更难 (参见练习 27). 这枚特殊的硬币将出现在检测树的 2 个叶子, 一个叶子代表这枚硬币太轻, 另一个叶子代表这枚硬币太重的情况. □

确定更复杂的搜索树的叶子数量和高度, 是计算机科学领域中称为算法分析的一个主要问题. 在 3.4 节, 我们要确定排序 n 项列表时所产生的搜索树的叶子数量

和高度. 在 7.2 节关于枚举的部分中, 我们将讨论其他搜索树的叶子数量与高度之间的递归关系. 本节的最后, 我们给出 n 个标签顶点上的不同无向树数量的公式. 每一个 3 标签树都是两条边的路径, 它们之间的差别是 3 个标签中的哪一个做中间顶点: 1-2-3, 1-3-2, 2-1-3(交换两个叶子的位置不产生不同的树). 尽管下面的公式是由 Cayley 首先给出证明的, 这里给出 Prufer 给出的更简单的证明.

定理 5(Cayley, 1889) *存在 n^{n-2} 个不同的 n 标签无向树.*

证明 观察发现, n^{n-2} 是长度为 $n-2$ 的 n 标签序列的数量. 现在, 我们在 n 个标签的树与长度为 $n-2$ 的 n 标签序列之间构造一个一一对应. 为了方便起见, 设标签是数字 $1, 2, 3, \cdots, n$.

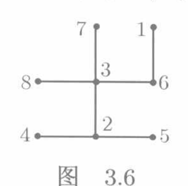


图 3.6

对于关于 n 个数的任意树, 如下构建一个长度为 $n-2$ 的序列 $(s_1, s_2, \cdots, s_{n-2})$. 设 l_1 是这棵树中编号最小的叶子, s_1 是与这个叶子相邻的顶点的号码. 对于图 3.6 中的树, 有最小号码的叶子是 1, 它的邻居顶点的号码是 6. 所以 $s_1 = 6$. 从这棵树中删除叶子 l_1, 并重复这一过程. 对于图 3.6 中的树, 删除 1 之后, 这棵树的最小号码叶子 l_2 是 4, 而它的邻居是 2. 所以 $s_2 = 2$. 持续这一过程, 我们得到 $l_3 = 5$, $s_3 = 2$, $l_4 = 2$, $s_4 = 3$, $l_5 = 6$, $s_5 = 3$, $l_6 = 7$, $s_6 = 3$. 当剩余树退化成由一条边连接的两个叶子时, 我们停止这一过程. 图 3.6 中 8 标签树的 6 标签序列是 (6, 2, 2, 3, 3, 3). 这样的序列称为 Prufer*序列*(Prufer sequence).

接下来证明, 任意这样的长度为 $(n-2)$ 的 n 项序列定义唯一一棵 n 项树. 我们只是把构建这一序列的过程反过来. 观察发现, 叶子 (度为 1 的顶点) 将永远不在这个序列中出现. 这个序列的第一个数是编号最小的叶子的邻居. 从刚才的观察可知, 这一最小编号叶子是不在这个序列中出现的最小数. 对于序列 (6, 2, 2, 3, 3, 3) 来说, 1 是不在这个序列中出现的最小数, 所以 1 是以 6 为邻居的叶子.

现在, 我们把这个最小叶子 (标号 1)(它是与这个序列的第一项相邻的叶子, 它在树中的位置已经确定) 放到一边, 而把第一项 (项 6) 看成一个叶子, 它与剩余序列中的某个项相邻. 然后, 我们重复这一过程, 识别由剩余的 $(n-3)$ 标签序列所指定的剩余 $(n-1)$ 标签树中的最小叶子. 对于剩余序列 (2, 2, 3, 3, 3), 标签 4 是不在这个序列中的剩余数中的最小数 (标签 1 已被删除), 所以项 4 是叶子, 它与剩余序列的第一个标签 2 相邻. 在更短的序列 (2, 3, 3, 3) 中继续, 标签 5 是其余数的最小叶子, 且它与标签 2 相邻. 现在, 标签 2 变成剩余序列 (3, 3, 3) 的一个可能的"叶子". 注意, 当前可能的叶子标签是 2, 6, 7, 8, 所以标签 2(最小可能叶子) 与项 3 相邻. 在序列 (3, 3) 中, 标签 6 与标签 3 相邻. 最后, 标签 7 与标签 3 相邻. 还剩下标签 3 和标签 8, 它们一定彼此相邻.

上面所述的由给定的 Prufer 序列出发构建标签树的过程适用于任意 Prufer 序列. 因此, n 标签树和长度为 $n-2$ 的序列之间的对应是一一对应, 从而证明了定理. ◇

练 习 3.1

练习概况 练习 3~18 给出关于树的理论. 练习 19~27 涉及各种使用树建模的问题.

1. 画出有下面各顶点数的所有非同构树:

(a) 4 个顶点; **(b)** 5 个顶点; **(c)** 6 个顶点.

2. 假设一个连通图有 20 条边. 顶点数的最大可能是多少?

3. 证明: 所有树都可以 2 着色.

4. 证明: 所有树都是可平面图.

5. 证明: 非有向连通图 G 是树 (即从根到每一个顶点存在唯一一条路径), 如果下面的条件之一成立:

(a) G 有 n 个顶点和 $n-1$ 条边; **(b)** G 的边数小于顶点数;

(c) 删除 G 的任意一条边将使 G 不连通.

6. 利用树是可平面图 (练习 4) 的事实和欧拉公式 (1.4 节定理 2) 重新证明定理 2.

7. 证明: 顶点个数大于 1 的任意树至少有两个度为 1 的顶点.

8. 证明定理 3 的推论的下面部分: **(a)**部分 (a); **(b)**部分 (b); **(c)**部分 (c).

9. 在构建从根到某个叶子的路径时, 通过计数每个内部顶点处的最大可能选择数目来重新证明: 对于任意高度为 h 的 m 元树, $l \leqslant m^h$.

10. 高度为 h 的 m 元树的最大顶点数是多少?

11. 证明: m 元树中内部顶点的比率大约是 $1/m$.

12. 对于给定的 h, 证明: 有 k 个叶子的 m 元树在它是平衡树时高度最小.

13. 对于 $n>2$, 有 n 个顶点的树中度为 1 的顶点数最多是多少? 最少是多少?

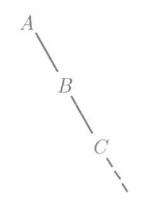
14. 证明: 对于一个不一定连通的图, 如果它没有回路, 但是 (在已有的两个顶点之间) 加入任意一条边总可以创建一条回路, 那么它是树.

15. 证明: 在有 n 个顶点的树中, (1.1 节例 5 中定义的) 最大独立集合的大小至少是 $n/2$.

16. 森林(forest) 是树的不相交并集构成的非连通图. 如果 G 是 t 棵树构成的有 n 个顶点的森林, 那么它有多少条边?

17. 证明: 二叉树中所有 l 个叶子的层次数之和至少是 $l\lceil\log_2 l\rceil$, 因此平均叶子层次数至少是 $\lceil\log_2 l\rceil$.

18. 设 T 是无向树. 如果选择顶点 x 为根时产生最小高度的根树, 那么 x 称为 T 的中心(center). 证明: 任意无向树至多有两个中心.



19. 考虑通过把各数值或部分和相加的方式对 n 个数求和的问题, 例如, $\{[(3+1)+(2+5)]+9\}$.

(a) 使用树表示这种相加过程. 内部顶点代表什么?

(b) 求 100 个数的和的 "相加树" 的最小可能高度是多少?

20. 右面的搜索树表示什么类型的搜索过程?

21. 可以用树表示二进制代码; 左分支是 0, 右分支是 1. 到一个字母 (顶点) 的路径是这个字母的二进制代码. 为了避免混淆, 有时候我们要求一个字母的代码的前缀不能是另一个字母的代码 (举例来说, 如果 K 被编码为 0101, 那么不能有编码为 0, 01, 010 的字母). 在这一要求下, 树中哪些顶点代表字母? 使用 n 位二进制序列可以编码多少字母?

22. 假设一次网球循环赛 (类似于例 1 中的二叉树循环赛) 中, 每名选手带来一罐新网球. 每场比赛用一罐球, 而另一罐则由这场比赛的胜者获得, 并用于下一轮比赛. 利用这一事实证明: 有 n 名选手的网球循环赛有 $n-1$ 场比赛.

23. 考虑有 32 名选手参加的一次网球循环赛 T(按图 3.2 中说明的树结构).

(a) 在前两轮比赛中淘汰多少名选手?

(b) 假设这一循环赛前两轮比赛的失利者有资格参加失利者循环赛 T'. T' 的前两轮比赛要淘汰多少选手? 注意, 所有网球循环赛都是平衡树; 第一轮比赛之后, 每一轮比赛参加的人数是 2 的幂.

(c) 假设在 T' 的前两轮比赛中失利的选手有资格参加另一个失利者循环赛 T''. T'' 的前两轮比赛中要淘汰多少选手?

(d) 假设在 T'' 的前两轮比赛中失利的选手有资格参加另一个失利者循环赛, 一直这样下去, 直到正好有一名总失利者 (最后一场比赛有两个人). 为了决定出总失利者, 需要多少次失利者循环赛?

24. 假设在一年的第一个星期有一个人发起书写连锁信的活动. 这封连锁信的每一位收信人都必须在下一周给另外 5 个人寄出相同的信. 6 周后, 这些连锁信共花费多少邮资 (每封信 40 美分)?

25. **(a)** 重复例 3, 现在假设只可以使用双向分支 (小于、大于或等于). 画出前 13 个字母的平衡搜索树, 并确定 n 字母搜索的高度.

(b) 假设对字母 A, B, C, D, E 的双向分支搜索树要利用如下所示的字母频率: A 20%, B 20%, C 30%, D 10%, E 20%. 构建一棵双向树, 使得识别一个字母所需的平均检测次数最小.

26. **(a)** 对至多包含一枚过轻硬币的 20 枚硬币, 重复例 4.

(b) 利用归纳法证明: 包括一枚过轻硬币的 3^n 枚或更少的硬币, 可以在至多 n 次称重后知道哪一枚硬币过轻.

27. 假设有 4 枚硬币, 其中可能有一枚硬币或者过轻或者过重 (所有 4 枚可能都是真的).

(a) 如果有另外一枚已知为真的硬币, 说明如何仅用两次称重确定 9 种可能情况中的哪一种情况成立.

(b) 证明: 如果没有额外的真币, 则仅仅两次称重是不够的.

28. 设 T 是有 n 个顶点的 m 元树, 且包含 i 个内部顶点和 l 个叶子. 假设 m 是偶数. 证明: n 总是奇数. 对于同样的 m 值, 给出两个 (小) 例子来说明 i 可能是偶数也可能是奇数.

29. 在定理 5 的证明中, 我们证明了 Prufer 序列 $(s_1, s_2, \cdots, s_{n-2})$ 唯一描述了有 n 个项的树. 对于下面各个 Prufer 序列, 构建相应的树.

(a) (4, 5, 6, 2) **(b)** (2, 8, 8, 3, 5, 4) **(c)** (3, 3, 3, 3, 3, 3)

3.2 搜索树和生成树

树为涉及一系列选择的问题的求解提供了自然框架, 无论是在图中搜索特定顶点, 还是寻找走出迷宫的道路, 或者为交通道路问题寻找最便宜解等, 都需要树. 前两章的大部分问题, 如同构、哈密顿回路、最小着色以及在街角设置警察等, 这些

问题的计算机解决方案都需要基于树的搜索. 通过设定持续进行的选择为根树的内部顶点, 而解决方案或 "死端" 为叶子, 可以组织搜索来寻找可能的解决方案. 无论是图内搜索, 还是迷宫内搜索, 还是为求最优解而在所有解空间搜索, 搜索过程最关注的是它是无遗漏的, 即, 保证检查所有可能情况.

本节给出 3 个涉及游戏的树搜索, 这里不涉及运筹学搜索的应用. 大部分运筹学树枚举问题涉及非常大的树, 并使用特殊的树 "剪裁" 算法. 作为这样的一个应用例子, 我们要在 3.3 节解决一个小型的旅行商问题. 第 4 章讨论 4 个隐式地使用树来搜索整个图的重要优化算法.

我们从在图内搜索开始. 对于很多应用, 我们需要使用图算法来检测一个图是否具有某种性质, 例如连通性或可平面性, 或者对诸如回路或完全子图这样的给定结构的所有出现进行计数. 这些算法通常利用生成树在图的各顶点和各边之间搜索来检测这些性质或结构. 图 G 的**生成树**(spanning tree) 是 G 的这样的子图, 它是包含 G 的所有顶点的树. 可以通过深度优先搜索或广度优先搜索来构造生成树.

为了构造深度优先生成树, 我们挑出某个顶点作为根, 然后开始用这个图的边构建从这个根开始的路径. 这条路径一直延续下去, 直到不重复使用已在树中的某个顶点就无法继续前进为止. 这条路径停止时的终端顶点是叶子. 现在, 我们返回这个叶子的双亲, 并尝试沿另一个方向从这个双亲开始构建一条路径. 当从这个双亲 y 出发的所有可能路径和它的其他孩子都已经构建起来时, 我们返回到 y 的双亲, 以此类推, 直到我们返回到根, 并已经检查了从根出发的所有其他可能路径.

为了构造广度优先生成树, 我们挑出某个顶点 x 作为根, 并把所有离开 x 的边 (连带这些边的端顶点) 放置在树上. 然后, 持续向这棵树中加入从与 x 相邻的顶点离开的边, 除非这样的边进入到的顶点已在这棵树上. 我们一层一层地持续这一过程.

注意, 如果这个图不是连通的, 那么不存在生成树. 因此有下面的结果.

检测无向图是否连通的算法

使用深度优先或广度优先搜索尝试构建生成树. 如果在搜索过程中能够达到这个图的所有顶点, 那么我们得到一棵生成树, 而且这个图是连通的. 如果搜索不能到达所有顶点, 那么这个图不是连通的.

深度优先搜索 (广度优先搜索) 可以达到连通图所有顶点的形式证明留作练习 (练习 9).

一个 (无向) 图的**邻接矩阵**(adjacency matrix) 是一个 (0, 1) 矩阵, 如果顶点 x_i 和 x_j 相邻则项 (i, j) 是 1, 否则项 (i, j) 是 0.

例 1 检测连通性

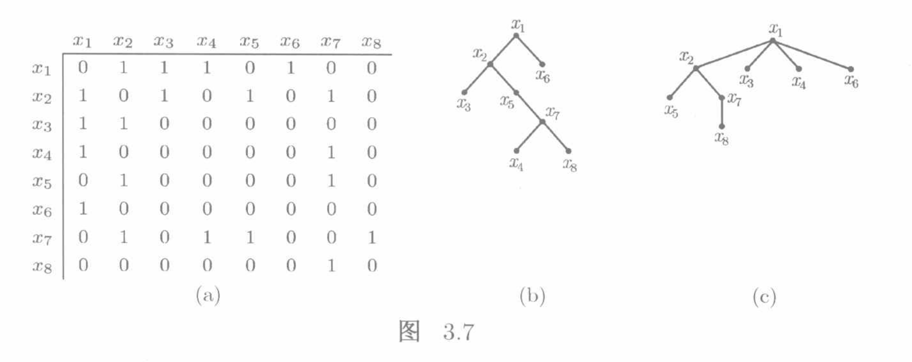
由图 3.7a 所示的邻接矩阵所表示的无向图 G 是连通的吗?

我们以 x_1 为根, 开始对 G 进行深度优先搜索. 对于每一个后继顶点, 我们选

择到达不在这棵树上的最低编号顶点的边作为这棵树的下一条边. 所以从 x_1 出发我们到达 x_2.

从 x_2 出发, 我们到达 x_3. 因为除了 x_1 和 x_2(它们都已经在树上) 之外, x_3 不与任何其他顶点相邻, 我们从 x_3 返回到 x_2, 然后继续从 x_2 开始搜索, 来到 x_5, 到达 x_7, 再到达 x_4. 在顶点 x_4, 我们返回到 x_7, 再到 x_8. 从 x_8, 我们必须一路返回到 x_1. 从 x_1, 我们到达 x_6. 这就完成这一搜索, 所有顶点都已经访问. 图 3.7b 给出所得到的生成树.

广度优先搜索的结果如图 3.7c 所示. □



	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
x_1	0	1	1	1	0	1	0	0
x_2	1	0	1	0	1	0	1	0
x_3	1	1	0	0	0	0	0	0
x_4	1	0	0	0	0	0	1	0
x_5	0	1	0	0	0	0	1	0
x_6	1	0	0	0	0	0	0	0
x_7	0	1	0	1	1	0	0	1
x_8	0	0	0	0	0	0	1	0

(a) (b) (c)

图 3.7

对图做深度优先搜索所需的计算时间与这个图中的边数成正比 (生成树中的每条边被访问两次, 不能使用的边正好尝试两次, 在其每一个端顶点尝试一次). 如果无向图是非连通的, 那么可以在任意一个顶点处利用深度优先搜索 (或广度优先搜索) 寻找它的连通分量; 然后再从一个不在前面的树上的顶点开始利用这个搜索来寻找另一个连通分量; 持续寻找其他连通分量, 直到所有顶点都已使用为止.

我们注意到广度优先搜索的一个重要性质. *广度优先生成树是由从根到图中其他每一个顶点的最短路径组成的*. 这一结论的证明留作练习 8. 如果我们只想寻找从根到图中某个特定顶点 x 的最短路径, 那么一旦到达 x, 深度优先搜索就可以停止 (不必构建整棵生成树).

现在, 我们把深度优先和广度优先搜索技术用于一些游戏. 在一个迷宫里, 顶点就是路径的交叉点. 在拼图游戏中, 顶点是这个拼图的不同拼块, 而边是可能的移动.

例 2 遍历迷宫

考虑图 3.8 所示的迷宫. 我们从一个标记为 S 的地方出发, 并要到达标记为 E 的一端.

我们使用深度优先搜索. 对于迷宫, 有一个构建深度优先搜索的便利规则 (其证明留作练习 17): 一直触摸迷宫的右侧墙壁. 当我们到达一个死胡同时, 沿着右侧墙壁到达墙头, 然后沿着左墙壁返回 (离开死胡同时这就是右墙壁). 当我们来到前面已经访问过的拐角时, 我们人为设置一个死胡同墙以便阻止我们实际到达那个

拐角 (如图 3.8 中的 S). 在图 3.8 的这一迷宫中, 我们使用实线表示 (向前) 构建的路径, 而点线表示反向返回的路径. 因为这个迷宫很容易直接搜索, 所以我们没有画出这一问题的搜索树 (在这一搜索树中, S 是根, 其他的拐角是内部顶点, 死胡同和 E 是叶子). □

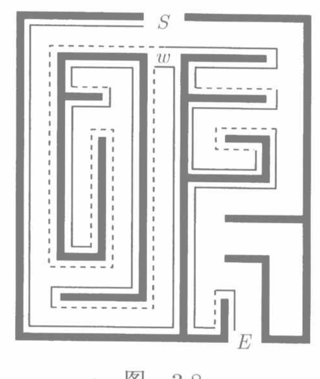


图 3.8

如果由可能路径表示的树很大, 那么广度优先方法就会变得很笨拙. 一次只沿着一条路径跟踪的深度优先更容易用纸笔跟踪或用程序跟踪. 另外, 当我们只需寻找各种可能解中的一个时, 它全力沿着一条路径寻找某个解, 而不是花大量时间构建大量的部分路径, 而最终只使用其中一条路径. 另一方面, 当我们想要的是包含最短路径的解或当存在非常长的死胡同路径 (而解路径相对较短) 时, 那么广度优先方法更好些. 第 4 章中的所有网络搜索算法使用广度优先搜索.

例 3 水罐灌水问题

假设我们有 3 个水罐, 其容量分别是 10 升, 7 升和 4 升. 开始时, 10 升的水罐是满的, 而其他 2 个是空的. 可以把水从一个水罐灌到另一个水罐中, 使得(或者被灌水的水罐变满, 或者灌水的水罐变空). 是否有方法在各水罐间灌水, 使得在 7 升或 4 升的水罐中得到 2 升水? 如果可以做到, 那么寻找最短灌水序列.

在这一枚举问题中, 位置或顶点是有序三元组 (a, b, c), 其中 a, b, c 分别是 10 升、7 升和 4 升水罐中的水量. 实际上, 只记录 (b, c) 就足够了, 即记录 7 升和 4 升罐中的水量, 因为 $a = 10 - b - c$. 有向边对应于从一个水罐到另一个水罐的灌水. 如图 3.9a 所示, 我们画出 a, b, c 坐标网格上的树. 这个坐标网以 $b = 7$, $c = 4$, $b + c = 10$ 为边界. 10 升水罐与 7 升水罐之间的灌水关系是一条水平边, 而 10 升水罐与 4 升水罐之间的灌水关系是一条垂直边, 7 升水罐与 4 升水罐之间的灌水关系是斜率为 -1 的对角线. 图 3.9b 给出这棵树的开始部分的标准树形式.

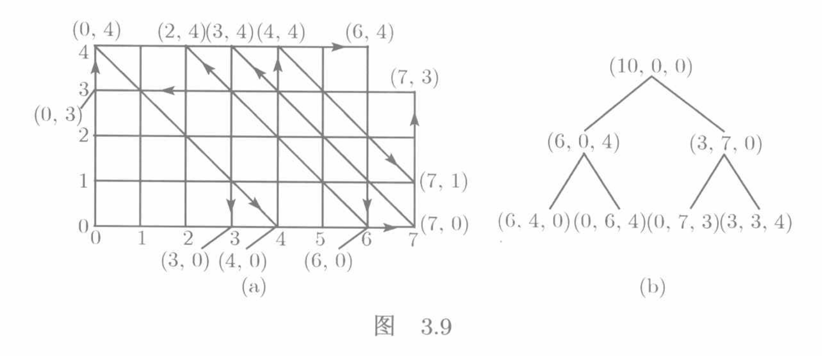


图 3.9

这棵搜索树的根是 (0, 0). 因为想要最短灌水序列 (即搜索树中的最短路径),

我们使用广度优先搜索. 从根开始, 我们能够到达位置 (7, 0) 和 (0, 4). 从 (7, 0) 开始, 我们能够到达新位置 (7, 3) 和 (3, 4); 而从 (0, 4) 开始, 我们能够到达新位置 (6, 4) 和 (4, 0). 至此所构建的树如图 3.9b 所示.

从位置 (7, 3) 开始, 唯一可到达的新位置是 (0, 3); 而从 (3, 4) 开始, 唯一可到达的新位置是 (3, 0). 从 (6, 4) 开始, 唯一可到达的新位置是 (6, 0); 从 (4, 0) 开始, 唯一可到达的新位置是 (4, 4). 到此, 我们已经检查了长度为 3 的所有路径. 此时唯一的新移动是从 (4, 4) 到 (7, 1) 和从 (6, 0) 到 (2, 4). 但是 (2, 4) 表示的是 7 升水罐中有 2 升水. 所以 (0, 0) 到 (0, 4) 到 (6, 4) 到 (6, 0) 到 (2, 4) 是得到 2 升水的灌水序列. □

例 4 传教士 — 野蛮人问题

假设 3 个传教士和 3 个野蛮人必须利用一艘可乘坐二人的小船过一条河. 说明如何完成这一过程, 使得任何时刻在河两岸野蛮人人数不超过传教士人数 (除非在岸上没有传教士).

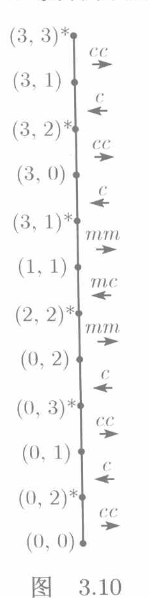


图 3.10

使用顶点表示当船在河岸一边时这个问题的状态. 我们利用有序对 (N_m, N_c) 标识每一个顶点, 其中 N_m 和 N_c 分别是邻近河岸的传教士人数和野蛮人人数. 当这条船在这一近处河岸时, 我们在标签后面放置一个星号. 我们从顶点 (3, 3)* 开始. 每一条边用一个箭头标识船的方向和船上的人: 用 1 个 m 表示船上的 1 名传教士, 用 1 个 c 表示 1 个野蛮人. 从 (3, 3)* 开始的初始合法移动只有两个野蛮人或者一名传教士和一个野蛮人过河的情况, 两种情况都有一个野蛮人留在远处的岸边, 另一个人把船划回来. 读者应该尝试着自己解决这一问题. 我们在图 3.10 中给出这个游戏的一个解, 这个解是第一次由两个野蛮人过河开始的. 关键步骤发生在第 6 次转移, 此时有两个人从远处河岸划回到近处河岸: 1 名传教士和 1 个野蛮人. □

假设我们构建了一棵树来为搜索或组织信息提供框架. 在这棵树中搜索特定顶点或处理信息通常涉及生成树上的深度优先遍历. 然而, 在深度优先遍历期间, 我们有若干次检验内部顶点的机会. 一棵树的**前序遍历**(preorder traversal) 是当在搜索中第一次遇到一个内部顶点时检查这个顶点的深度优先搜索. **后序遍历**(postorder traversal) 是当最后一次遇到一个内部顶点时 (在搜索从这个顶点和它的子树返回之前), 检查这个内部顶点. 如果树是二叉的, 那么可以定义**中序遍历**(inorder traversal), 它在内部顶点的左子树和右子树的遍历之间检查这个顶点. 图 3.11a 和图 3.11b 分别展示了根据前序遍历和后序遍历对图 3.11a 中的树的顶点的编号. (在每个内部顶点处, 我们假设在右孩子之前访问左孩子.)

前序遍历和后序遍历的一个重要性质是: 在前序遍历中, 一个顶点领先它的孩

子和所有其他后代, 而在后序遍历中, 一个顶点跟随在它的孩子和它的其他所有后代之后.

现在, 我们给出每种遍历的例子. 对于词典查找的二叉树搜索例子 (3.1 节的例 2), 顶点的字母顺序对应于中序遍历. 在遍历一棵树寻找有特殊标签的顶点时, 我们应该检测深度优先搜索中遇到的每一个新顶点, 看它是否有所求的标签. 推迟这样的检测不会带来任何好处. 因此在搜索指定顶点时, 我们应该根据前序遍历检查顶点.

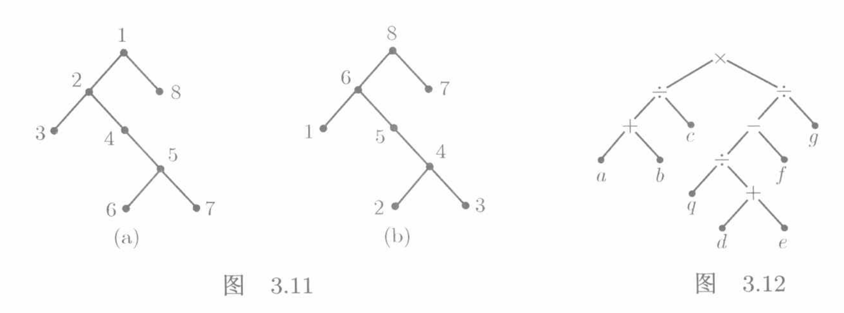


图 3.11

图 3.12

我们用一棵算术树说明对后序遍历的需求. 算术表达式:

$$((a+b)\div c)\times(((q\div(d+e))-f)\div g)$$

可以被分解成如图 3.12 所示的二叉树. 内部顶点是算术运算符, 叶子是变量. 为了求出这个算术表达式, 我们需要根据这个算术树的后序遍历来执行内部顶点所指定的这些运算, 即只有在一个运算符顶点的两棵子树所代表的子表达式被求出之后, 我们才能执行这个运算.

练 习 3.2

练习概况 前 6 道练习是关于构建生成树的练习, 接下来的 5 道练习是关于生成树性质的练习. 练习 12~25 基于本节的例子. 练习 26~27 是关于遍历的练习.

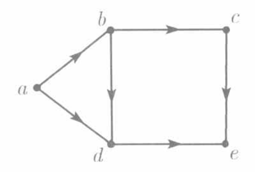
1. 对于下面各图, 寻找深度优先生成树:

(a) K_8(8 顶点完全图); **(b)** 2.2 节图 2.5; **(c)** 2.3 节图 2.10; **(d)** $K_{3,3}$ (完全二部图).

2. 对于练习 1 中的每一个图, 寻找广度优先生成树.

3. 对于下面各图, 寻找 (关于同构的) 所有生成树:

(a) 图 2.4; **(b)** K_4; **(c)**



(d) $K_{3,3}$(完全二部图).

4. 如果一个图有如下所示的邻接矩阵, 检查它是否连通.

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
x_1	0	0	1	0	0	1	0	1
x_2	0	0	1	0	1	0	1	0
x_3	1	1	0	1	0	0	0	0
x_4	0	0	1	0	1	1	1	0
x_5	0	1	0	1	0	0	0	1
x_6	1	0	0	1	0	0	0	0
x_7	0	1	0	1	0	0	0	1
x_8	1	0	0	0	1	0	1	0

5. 考虑有 25 个顶点 $x_2, x_3, \cdots, x_{26}$, 且当且仅当整数 i 和 j 有公约数时有边 (x_i, x_j) 的无向图. 这个图有多少个连通分量? 为每一个连通分量构建生成树.

6. 证明: 对于有 n 顶点的图, 不形成回路的 $n-1$ 条边组成一棵生成树.

7. 证明: 只有一棵生成树的连通无向图是树.

8. 证明: 广度优先生成树包含从根到其他每一个顶点的最短路径.

9. **(a)** 证明: 对于无向连通图, 深度优先搜索到达所有顶点.

(b) 对于广度优先搜索, 重做问题 (a).

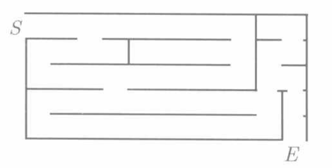
10. **(a)** 证明: 对于图 G, 从给定根 a 开始的任意深度优先生成树的高度一定至少与以 a 为根的广度优先生成树的高度相同.

(b) 如果从不同的根开始, 图 G 的深度优先生成树的高度有可能小于图 G 的广度优先生成树的高度. 寻找符合这一情况的图.

11. 割集是连通图 G 的这样一个边集合 S, 删除 S 将会断开 G, 但删除 S 的任意真子集都不破坏 G 的连通性. 证明: G 的任意割集与 G 的任意生成树至少有一条公共边.

12. **(a)** 使用深度优先搜索, 寻找右面迷宫中从 S 到 E 的路线.

(b) 有多少条穿过这个迷宫的 (无循环及后退的) 路线?



13. 使用深度优先搜索重复例 3.

14. 对图 3.8 执行广度优先搜索, 寻找从 S 到 E 的最短路径.

15. 对于例 3, 寻找在一个水罐里得到 2 升水的另一条 (实际上更短的) 路径.

16. **(a)** 对容量分别为 8, 5, 3 的水罐及在一个水罐中得到 4 升水的目标, 重复例 3.

(b) 对水容量分别为 12, 8, 5 的水罐及在一个水罐中得到 2 升水的目标, 重复例 3.

17. 证明: 右侧贴墙的规则总使我们可以走出迷宫.

18. 对于例 3, 使用深度优先搜索证明: 对于 0~10 的任意升水量, 总可以在某个水罐中得到它.

19. 假设我们用一只小船把一条狗、一只山羊和一袋子罐头运到河对岸, 每次只能运送三者之一过河 (连同摆渡人). 设狗和山羊不能单独留在岸上 (当摆渡的人不在时), 也不能让山羊和罐头单独留在岸上, 寻找一个方案使得它们都可以过河.

20. 3 个生性多疑的妇人和她们的丈夫要过河. 这一行人必须用一只最多只能容纳两个人的小船过河 (从近河岸到远河岸). 任何时候都不能让某位丈夫与别人的妻子单独留在河岸上, 在这样的条件下寻找使得这 6 个人过河的小船行程序列.

21. 如果可能的话, 这次有 4 位生性多疑的妇人和她们的丈夫要过河, 重复练习 20.

22. 重做练习 20, 这一次是 4 位妇人和她们的丈夫, 以及最多可承载 3 人的小船.

23. 在下面的附加条件下重复例 4: 野蛮人中只有 1 个人可以划船 (所有传教士都可以划船).

24. 下面是水罐灌水问题的一个变形. 沙漏是一种时钟, 当上半部分的沙子经过狭窄的开口流空到下半部分时, 要经过一段特定的时间. 如果把一个沙漏设计成上面的沙子流到下面需要花 10 分钟, 而且比如说 7 分钟后, 把我们这个沙漏反转过来, 那么沙子将倒流, 而且将再花 7 分钟时间使沙子流空. 假设你有两个沙漏, 一个是 4 分钟沙漏, 一个是 7 分钟沙漏. 描述一种方案: 从这两个沙漏开始, 并上下反转它们以便测量出 9 分钟的时间段. (提示: 在开始的时候, 你从一个沙漏开始或者两个沙漏同时开始, 然后当一个沙漏流空时, 决定或者令它重新开始, 或者开始或反转另一个沙漏.)

25. 一组 4 个人必须通过一座桥, 且每次只允许两个人携伴而行. 此时是晚间, 且只有一只手电筒. 过桥的一行人, 或者是一个人或者是两个人, 必须随身带着手电筒. (这个手电筒必须被拿来拿去, 不能扔掉它.) 寻找在 17 分钟内 4 个人都到达桥的另一端的方法, 已知 A 过桥需要 1 分钟, B 过桥需要 2 分钟, C 过桥需要 5 分钟, D 过桥需要 10 分钟. 当两个人过桥时, 他们必须以较慢者的速度行走.

26. 对于下面各图, 以前序遍历和后序遍历的顺序列出它们的顶点:

(a) 图 3.1a; **(b)** 图 3.1b; **(c)** 图 3.4.

27. 扩展图 3.12 的算术树, 使其包含一元运算符, 如逆或 sin(). 给出下面表达式的树:

$$\sin(((a+((b\times c)^{-1}+((a+d)\times e)))-(c+e))+(a-b)^{-1})$$

28. 设 T 是连通无向图 G 的生成树. 证明: 把任意不在树中的边加到 T 中将产生唯一一条回路.

29. 证明: 对于图 G 的任意生成树 T' 和 T'', 可以通过生成树序列 $T_1, T_2, \cdots, T_m$ 把 T' 转化成 T'', 其中 $T'=T_1$, $T''=T_m$, T_k 是通过把 T_{k-1} 在 T' 中的一条边去掉并向 T_{k-1} 中加入 T'' 的一条边得到的 (m 是 T' 与 T'' 不同的边的边数). (提示: 利用练习 28 并对 m 使用归纳法.)

30. 假设在二叉树 T 的前序遍历过程中, 对遍历的每一个内部顶点写下一个 1, 而对每一个叶子写下一个 0, 从而构建一个 1 和 0 的序列. 如果 T 有 n 个叶子, 那么这个序列将有 n 个 0 和 $n-1$ 个 1. 我们称这一序列是 T 的特征序列 (characteristic sequence). (这样的序列唯一确定一棵树.)

(a) 寻找有特征序列 110100110100100 的二叉树.

(b) 证明: 任意特征序列的最后两位数字是 0(假设 $n \geqslant 2$).

(c) 对于正整数 n, 证明有 n 个 0 和 $n-1$ 个 1 的二进制序列是某个二叉树的特征序列, 当且仅当对于 $1 \leqslant k \leqslant 2n-2$, 这个序列的前 k 个数字中 1 的数目至少与 0 的数目同样多.

31. **(a)** 编写从图的邻接矩阵出发构建深度优先生成树的程序.

(b) 对于广度优先, 重复部分 (a).

32. 使用广度优先逆向生成树 T(边都是由孩子指向双亲), 如下在存在欧拉回路的有向图中构建欧拉圈. 从树 T 的根开始, 绘出一条路径, 使得在任意顶点 x 处, 只有当没有其他离开 x 的未使用边, 才选择 T 中的一条边到达 x(在 T 中) 的双亲. 验证这个算法的正确性. 编写这个算法的程序.

3.3 旅行商问题

本节说明树在运筹学图优化问题中的应用. 旅行商问题是为遍访一组城市并返回原地的旅行商寻找最小成本路线. 我们要在有相关成本矩阵 C 的完全图中寻找最小成本哈密顿回路. C 中的项 c_{ij} 是使用从第 i 个顶点到第 j 个顶点的边的成本. 旅行商问题在运筹学中以很多不同的形式出现, 是著名的难题 (它是一个 NP 完全问题, 参见附录 A.5). 其中一个例子是在印刷电路板的指定位置钻孔时自动钻台的移动方案.

我们给出两种解决旅行商问题的方法. 这两种方法都使用树, 但它们采用两种非常不同的方式. 第一种方法是在搜索最廉价的哈密顿回路时, 系统地考虑构建哈密顿回路的所有可能方法. 这一方法使用一个"分支定界"方法来限制最小解搜索中必须检查的搜索树上的不同顶点的数量. 树上的每一个顶点将代表一个部分哈密顿回路, 每一个叶子是一个完整的哈密顿回路. 注意, n 顶点完全图有 $(n-1)!$ 个不同的哈密顿回路. 例如, 50 个城市 (顶点) 的旅行商问题有 $49! \approx 6\times10^{62}$ 个可能回路. 尽管这种"分支定界"方法还不足以应用于实际解决大型旅行商问题, 但是它的确大幅减少了必须检查的回路数量.

这导致另一种策略, 即我们的第二种方法, 这个方法是使用一种相当快的算法来构建一个接近最小的解, 对于像旅行商问题这样包含非常巨大的树枚举的问题, 寻找精确解需要花费大量的时间, 因此促使研究人员把精力放在这类问题的启发式、近似最优的算法上.

注意, 成本矩阵中的数值本身可能就是一个估测值, 例如, 卡车所跑的时间, 所以使用这些数值得到任意解也只能是一种估测. 寻找任意图的最小着色, 或检测同构性同样是拥有大量可能解的, 以搜索树为基础的问题. 近似最小着色算法已经得到广泛的关注, 但遗憾的是, "近似同构" 通常没有价值.

考虑有 4 个顶点 x_1, x_2, x_3, x_4 的小型旅行商问题. 设这一问题的成本矩阵是图 3.13 中的矩阵. 项 c_{ij} 是从顶点 (城市)i 到顶点 j 的成本. (注意, 不要求 $c_{ij} = c_{ji}$.) 对角线上的无穷大成本表示我们不能使用这些项. 哈密顿回路使用成本矩阵 C 中的 4 个项, 每一行和每一列上各有一项, 使得项 (边) 的任意真子集都不形成子回路. 后面的限制表明, 如果 (比如说) 选择了项 c_{23}, 那么就不能再使用 c_{32}, 因为这两项形成一个长度为 2 的子回路. 类似地, 如果使用了项 c_{23} 和 c_{31}, 那么就不能使用项 c_{12}.

我们首先说明如何得到这个旅行商问题的成本下界. 因为每个解必须包含一个第一行的项, 如果我们从成本矩阵的第一行的每一项减去一个常量, 那么最小行程的边不发生变化 (当然最小行程成本将因这个常量而发生变化). 我们在不使第一行的任意数值变为负数的前提下, 从这一行中减去尽可能大的数, 即, 减去第一

行的最小值 3. 对于其他 3 行也这样做. 图 3.14 给出了修改后的成本矩阵.

从 \ 到	1	2	3	4
1	∞	3	9	7
2	3	∞	6	5
3	5	6	∞	6
4	9	7	4	∞

图 3.13

从 \ 到	1	2	3	4
1	∞	0	6	4
2	0	∞	3	2
3	0	1	∞	1
4	5	3	0	∞

图 3.14

现在, 图 3.14 中的所有行都包含一个 0 项. 从各行减去的总数是 3+3+5+4=15, 使用图 3.14 中的成本数据的最小行程将比使用图 3.13 的原来的成本数据少花费 15. 而且, 修改后的问题的最小行程的边与原始问题的最小行程边相同.

使用类似的形式, 可以从任意列减去一个常量而不改变最小行程的边集合. 因为我们想要避免使任意项为负, 所以考虑只从所有项都为正数的列中减去一个常量. 对于图 3.14, 这样的列只有列 4, 它的最小值是 1. 所以我们从图 3.14 的最后一列减去 1, 得到图 3.15 的矩阵. 此时图 3.15 的每一行和每一列都有一个 0 项.

从 \ 到	1	2	3	4	
1	∞	0	6	3	下界=16
2	0	∞	3	1	
3	0	1	∞	0	
4	5	3	0	∞	

图 3.15

使用图 3.15 的最小行程的成本比使用图 3.13 的原始成本减少 15+1=16. 可以使用这种成本的减少量得到最小行程成本的下界: 使用图 3.15 的成本的最小行程的花费至少是 0, 因此使用图 3.13 的最小行程的花费一定至少是 16. *一般地, 旅行商问题的下界等于把原始成本矩阵变成每一行和每一列都有 0 项的新成本矩阵时需要减去的常量之和.*

现在, 我们已经为"分支定界"方法的分支部分做好了准备. 看一下图 3.15 中等于 0 的项. 比如说 c_{12}. 我们或者使用 c_{12}, 或者不使用 c_{12}. 我们在这一选择处进行"分支". 对于不使用 c_{12} 的情况, 我们通过设 $c_{12} = \infty$ 来表示不使用 c_{12} 的选择. 修改后的图 3.15 的第一行的最小值此时是 c_{14}=3, 所以我们从第一行减去这个量 (这一行的所有项都减去此行的最小值 3). 类似地, 可以从第二列减去 1. 如果我们不使用项 c_{12}, 那么我们得到图 3.16 的新成本矩阵. 因此, 不使用 c_{12} 的原始问题 (图 3.13) 的任意行程的花费至少是 16+(3+1)=20.

如果使用 c_{12} 构建图 3.15 的行程, 那么这个行程的其余部分不能使用第一行或第二列中的其他项, 而且同样必须设置 c_{21} 等于 ∞(为了避免子回路 $x_1 - x_2 - x_1$). 简化后的矩阵在第二行的最小值是 1, 所以我们从第二行减去 1 得到图 3.17 的成本矩阵. 因此, 使用 c_{12} 的图 3.17 的行程的下界是 16+1=17.

因为得到的下界 17 小于不使用 c_{12} 时的下界 20, 所以考虑使用 c_{12} 的行程, 分

别确定使用和不使用第二个等于 0 的项时的下界. 只要下界小于等于 20, 我们就继续扩展这些使用 c_{12} 的部分行程. 如果下界超过 20, 那么我们不得不返回, 考虑不使用 c_{12} 的部分行程. 这一问题的选择树是二叉树, 其内部顶点代表使用 c_{ij} 还是不使用 c_{ij} 这样的选择 (参见图 3.19). 在任意阶段, 只要使用 c_{ij} 的部分行程的下界小于不使用 c_{ij} 的行程的下界, 我们就无需查看不使用 c_{ij} 的可能行程的子树.

	到	1	2	3	4	不使用 c_{12}
从	1	∞	∞	3	0	下界=20
	2	0	∞	3	1	
	3	0	0	∞	0	
	4	5	2	0	∞	

图 3.16

	到	1	3	4	使用 c_{12}
从	2	∞	2	0	下界=17
	3	0	∞	0	
	4	5	0	∞	

图 3.17

考虑其他有值 0 的项来扩展使用 c_{12} 行程. 下一个 0 项无需与 c_{12} 连通, 即它无需是形如 c_{i1} 或 c_{2j} 的项, 但是为了方便起见, 我们挑出第二行中的项. 图 3.17 中第二行的唯一 0 项是 c_{24}.

	到	1	3	使用 c_{12} 和 c_{24}
从	3	0	∞	下界=17
	4	∞	0	

图 3.18

同样, 可以选择使用 c_{24} 或者不使用 c_{24}. 不使用 c_{24} 将使下界增加 2(我们设置 $c_{24}=\infty$ 之后, 2 是第二行的最小项, 列 4 仍然有一个 0). 使用 c_{24} 将不增加下界, 所以我们进一步沿着 c_{12} 扩展使用 c_{24} 的部分行程. 同样, 我们删除图 3.17 中的第二行和第四列, 并设置 $c_{41}=\infty$(来阻断子回路 $x_1-x_2-x_4-x_1$). 图 3.18 给出新的剩余成本矩阵 (所有行和列仍有 0 项).

现在, 完成这一行程的方法是显然的: 使用项 c_{43} 和 c_{31} 形成一个完整的行程 $x_1-x_2-x_4-x_3-x_1$. 实际上我们没有其他选择, 因为不使用 c_{43} 或 c_{31} 迫使我们使用一个 ∞(这代表一条被禁用的边). 因为, $c_{43}=c_{31}=0$, 所以这一行程有一个等于图 3.18 中的下界 17 的成本. 另外, 这一行程一定是最小的, 因为它的成本等于我们的下界. 利用图 3.13 中的原始成本矩阵, 重新检查这一行程的成本, 我们有 $c_{12}+c_{24}+c_{43}+c_{31}=3+5+4+5=17$.

我们用图 3.19 中的决策树对前面所做的推理加以概括, 它给出了我们所做的选择以及各选择的下界.

应该对如何充分利用分支定界技术做出一般性的概括. 在每一个阶段, 我们都要选择进行分支的下一项 (使用还是不使用这一项), 也就是 0 项, 它的删除将使下界的增量最大. 在图 3.15 中, 对所有 0 项的检查表明, 不使用项 c_{43} 将使下界增加 3+3=6(3 是第 4 行和第 3 列中新的最小值). 所以, 从理论上看, 对于第一个分支来说, c_{43} 比 c_{12} 好, 因为不使用 c_{43} 的行程的子树具有更大的下界, 使得我们无需检

查那个子树中的可能行程的可能性更大.

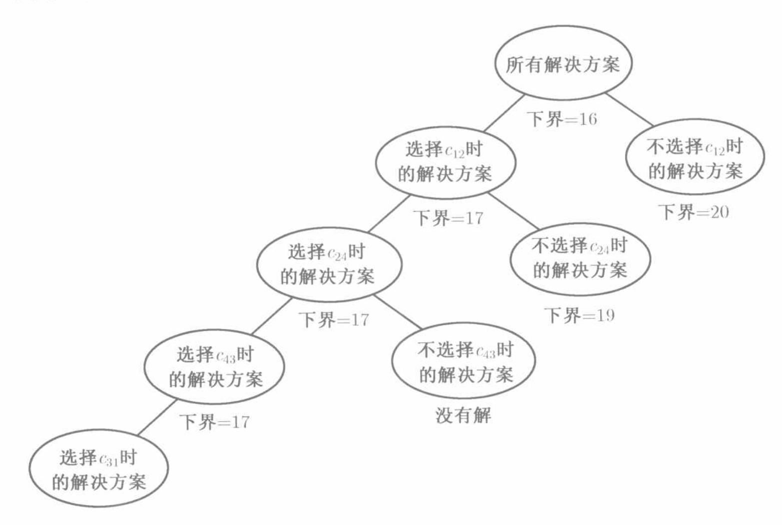


图 3.19

现在, 考虑图 3.20 中的 6×6 成本矩阵. 我们在图 3.21 中给出这个成本矩阵的前几次分支. 我们圈出在每一个阶段进行分支的 0 项.

从 \ 到	1	2	3	4	5	6
1	∞	3	4	2	8	3
2	5	∞	3	4	4	5
3	4	1	∞	5	3	4
4	4	2	6	∞	4	5
5	3	3	3	5	∞	4
6	7	4	5	6	7	∞

图 3.20

分支定界法不仅用于解决优化问题. 它还是人工智能的核心工具. 会玩象棋的计算机 (程序) 制作下几步可能棋的树, 然后再给所产生的棋盘局面分配某种类型的 "值". 产生坏局面的一步棋得到非常高的值, 从而不再跟踪从这一步开始的棋局. 在更有希望的局面处, 考虑所有合理的下一步棋, 而且对它们所产生的局势进行评估. 计算机越强大, 它就能够检查更多步后继局面, 当然, 与计算机的速度相比, 评估局面的过程也同样重要.

当成本对称 (即 $c_{ij}=c_{ji}$) 且满足三角形不等式 $c_{ik}\leqslant c_{ij}+c_{jk}$ 时, 我们可以给出下面获得 (近似最小的) 好行程的快速算法. 大多数旅行商问题都满足这两条假设. 在给出算法并举出其使用例子之后, 我们将证明在最坏情况下, 这个近似算法的行程总小于真实最小行程的两倍. 真实最小值两倍的定界听起来也许不好, 但是, 对于许多需要估计大致数值 (而且如果需要, 可以以后再计算真实的最小行程) 的情况, 这样的定界是完全可以接受的. 另外, 在实践中, 我们的近似算法可以找到接近于真实最小值的行程. 这一算法使用逐次最近邻居策略.

近似旅行商行程的构造

(1) 挑选出任一顶点作为由 1 个顶点组成的起始回路 C_1.

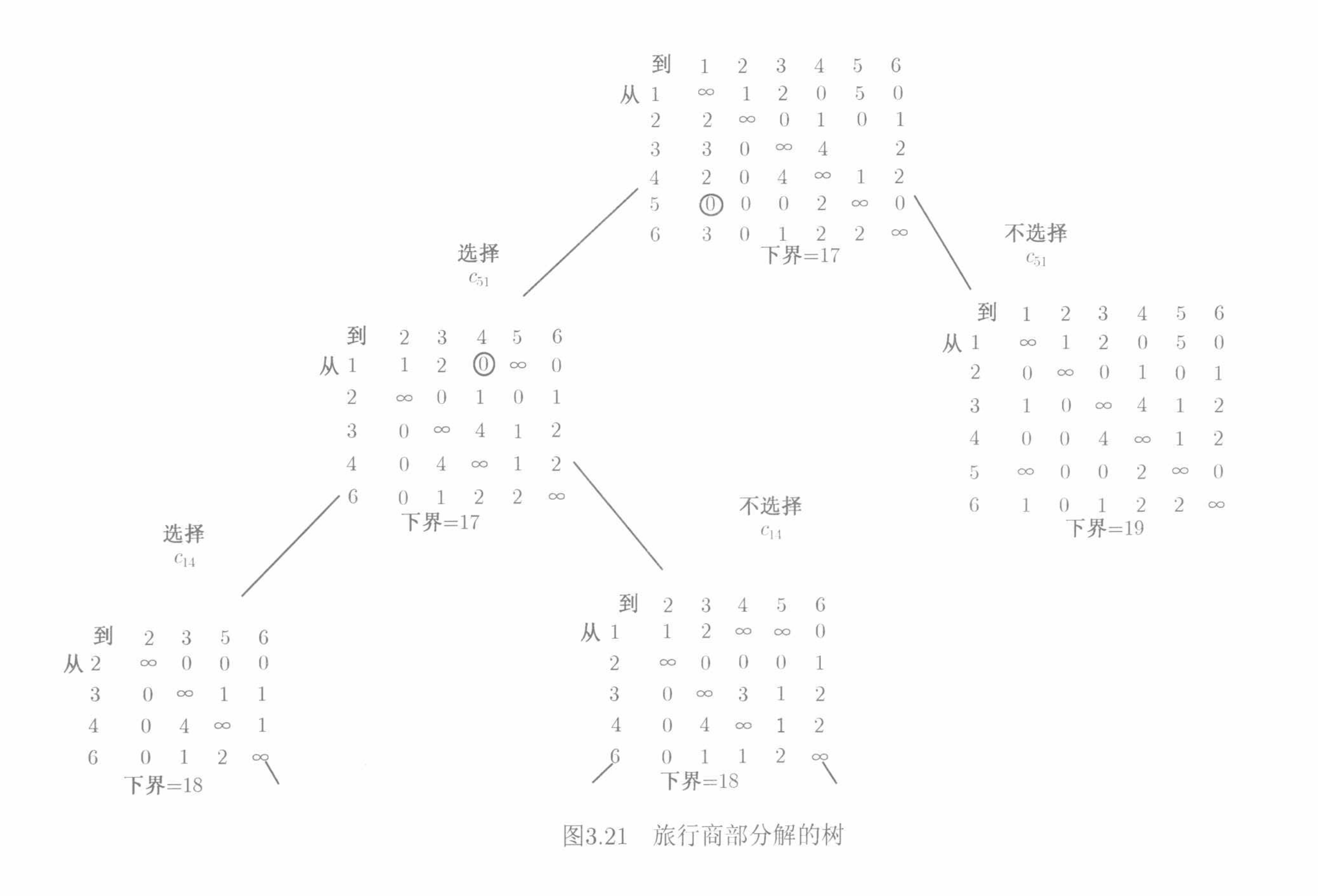


图3.21 旅行商部分解的树

(2) 给定 k 顶点回路 C_k, $k \geqslant 1$, 寻找不在 C_k 上且与 C_k 上的顶点 y_k 最近的顶点 z_k[①].

(3) 设 C_{k+1} 是把 z_k 插入到 C_k 中 y_k 的前面而得到的 $k+1$ 顶点回路.

(4) 重复步骤 (2) 和 (3), 直到形成一个 (包含所有顶点的) 哈密顿回路.

例 1 近似旅行商行程

从\到	1	2	3	4	5	6
1	∞	3	3	2	7	3
2	3	∞	3	4	5	5
3	3	3	∞	1	4	4
4	2	4	1	∞	5	5
5	7	5	4	5	∞	4
6	3	5	4	5	4	∞

图 3.22

我们把上面的算法运用于成本矩阵由图 3.22 所示的 6 顶点旅行商问题. 命名顶点 x_1, x_2, x_3, x_4, x_5, x_6. 我们从 C_1 等于 x_1 开始. 顶点 x_4 距离 x_1 最近, 所以 C_2 是 $x_1-x_4-x_1$. 顶点 x_3 是不在 C_2 上的距离 C_2 上的顶点最近的顶点, 即这个顶点离 x_4 最近, 因此 $C_3 = x_1 - x_3 - x_4 - x_1$. 现在有两个顶点 x_2 和 x_6, 它们距离 C_3 中的顶点都是 3 个单位. 假设我们挑选 x_2. 把它插入到 x_3 的前面得到 $C_4 = x_1 - x_2 - x_3 - x_4 - x_1$. 顶点 x_6 距离 x_1 还是 3 个单位, 所以我们在 x_1 之前插入 x_6, 得到 $C_5 = x_1 - x_2 - x_3 - x_4 - x_6 - x_1$. 最后, x_5 距离 x_3 和 x_6 为 4 个单位. 在 x_6 之前插入 x_5, 得到近似最小行程 $C_6 = x_1 - x_2 - x_3 - x_4 - x_5 - x_6 - x_1$, 我们计算它的成本等于 19.

对于这种情况, 利用近似行程构造算法得到的这一行程的长度相当接近最小值 (最小值等于 18). 近似行程的长度通常依赖于起始顶点. 如果我们怀疑近似行程不那样接近最优长度, 那么通过尝试以其他顶点为起始顶点 (形成 C_1) 并运用算法, 我们将得到另一个近似最小行程. 取由近似行程构造法生成的行程集合中最短的行程将给出对真实最小行程的改进评估. □

定理 由近似行程构造法生成的行程成本小于最小旅行商成本的两倍.

证明 (选读)

假设我们依据近似行程构造法依次构建 k 顶点回路 C_k. 设 S_k 是把 C_k 中的顶点与 G 中其他顶点相连的真实最小行程 C^* 中的边的子集 (下面将精确地描述 S_k). 当 $k=1$ 时, 设 S_1 是由 $C^* - e^*$ 组成的, 其中 e^* 是 C^* 中成本最高的边. 因为 C_1 是单一顶点, 且 $C^* - e^*$ 是一条哈密顿路径, 所以 $S_1 = C^* - e^*$ 把 G 的其余部分与 C_1 相连.

为了更加具体, 观察图 3.23a 中的 8 顶点图, 其中 C_1 是顶点 x_6, 而 $S_1 = C^* - e^*$ 是实线边的哈密顿路径. 如果 $z_1 = x_3$, 因而 C_2 是 $x_6 - x_3 - x_6$, 那么设 $S_2 = S_1 - (x_5, x_6)$(C_2 由图 3.23b 中的虚线边表示, S_2 由实线边表示). 一般地, 我们是这样从 S_k 得到 S_{k+1} 的: 假设是通过把 z_k 插入到 C_k 上的顶点 y_k 的后面得到 C_{k+1} 的, 从 S_k 的路径中删除从 y_k 到 z_k 的子路径上的第一条边[②]. 在图 3.23c 中, 假设 $z_2 = x_8, y_2 = x_6$. 所以把 x_8 在 x_6 与 x_3 之间插入到 C_2 中, 从而得到 $C_3 = x_6 - x_8 - x_3 - x_6$. 从 S_2 删除边 (x_6, x_7) 得到 S_3, 因为 (x_6, x_7) 是 S_2

① 这里指的是, 对于 C_k 上的任意顶点 y 和不在 C_k 上的任意顶点 z, y 和 z 的距离都不小于 y_k 和 z_k 的距离. —— 译者注

② 原书描述不十分精确, 我们做了重写. —— 译者注

中从 x_6 到 x_8 的路径上的第一条边.

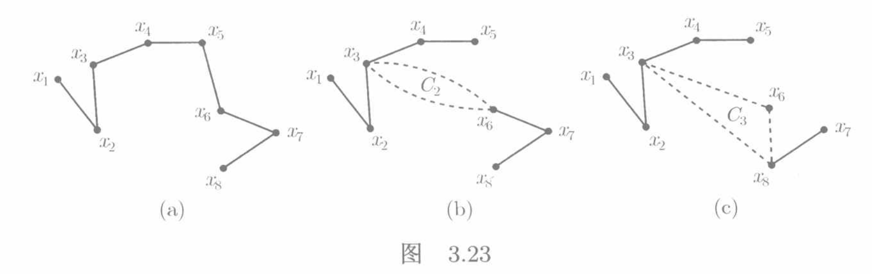


图 3.23

根据选出 $z_2 = x_8$, $y_2 = x_6$ 的最短边法则, 我们知道 c_{68}[边 (x_6, x_8) 的成本] 是 $C_2(= x_6 - x_3 - x_6)$ 与 G 的其余部分之间的所有边中成本最小的. 因此 $c_{68} \leqslant c_{67}$[其中 (x_6, x_7) 是从 S_2 中删除得到 S_3 的边]. 使用这一事实和三角不等式, 我们证明把 x_8 插入到 C_2 得到 C_3 有成本净增量 $\leqslant 2c_{67}$. C_3 相对于 C_2 的成本增加是 $c_{68} + c_{38} - c_{36}$, 因为边 (x_6, x_8) 和 (x_3, x_8) 取代了边 (x_3, x_6). 但是根据三角不等式, $c_{38} \leqslant c_{36} + c_{68}$ 或者 $c_{38} - c_{36} \leqslant c_{68}$. 因此 $c_{68} + (c_{38} - c_{36}) \leqslant c_{68} + c_{68} \leqslant 2c_{67}$.

在把每一个后继的 z_k 插入到 C_k 时, 可以运用相同的讨论来证明 C_{k+1} 相对于 C_k 的成本增加至多等于从 S_k 到 S_{k+1} 的删除边的成本的两倍. 从 C_1 开始, 重复这一插入成本边界, 我们看到最后的近似哈密顿回路 C_n 的上确界是 $S_1 = C^* - e^*$ 中各边的成本之和的两倍, 而这个成本之和小于最小旅行商行程 $C*$ 的成本的两倍. ◇

练 习 3.3

1. 用 10 减去图 3.13 中每一个非对角线项的值. 现在, 对这个新的成本矩阵求解旅行商问题.
2. 对图 3.20 的成本矩阵求解旅行商问题.
3. 编写一个求解 6 城市旅行商问题的计算机程序, 并对图 3.22 的成本矩阵运用这个程序.
4. 使用专门推理证明图 3.22 的成本矩阵的最小行程成本是 18.
5. 一家塑料工厂每个月必须制作 5 种不同类型的塑料玩具. 从玩具 i 的生产到玩具 j 的生产转换成本是 c_{ij}, 如下面的矩阵所示. 寻找能够使 (随后的几个月里使用的) 月转化成本之和最小的玩具生产序列.

	到	T_1	T_2	T_3	T_4	T_5
从	T_1	—	3	2	4	3
	T_2	4	—	4	5	5
	T_3	5	3	—	4	4
	T_4	3	5	1	—	6
	T_5	5	4	2	3	—

6. 分配问题(assignment problem) 是这样一个匹配问题, 它有 n 名员工和 n 项工作以及一个成本矩阵, 其中成本矩阵的项 c_{ij} 表示分配员工 i 去做工作 j 的 "成本". 我们的目标是寻找能够使成本之和最小的员工与工作之间的一一匹配. 设图 3.13 中的所有对角线项等

于 5, 并利用分支定界法解决这个 4×4 的分配问题. 分配问题与旅行商问题有什么不同?

7. 使用近似算法, 寻找下面各成本矩阵的近似旅行商行程 (只使用主对角线上方的项):

(a) 图 3.13; **(b)** 图 3.20; **(c)** 练习 5.

8. 考虑下面构建旅行商近似行程的规则. 从单一顶点行程 T_1 开始, 依次加入这样的顶点: 把它加入到 T_k 形成 T_{k+1} 能够使成本增量最小, 即, 如果 x 被插入到 x_k 和 x_{k+1} 之间, 那么对所有 x_r 和 x_k 的选择, $c_{kr}+c_{r(k+1)}-c_{k(k+1)}$ 最小.

(a) 你能够证明上面方法的上界类似于近似算法所发现的上界吗? (使用相同的假设)

(b) 对 (i) 图 3.20(只使用主对角线上方的项) 和 (ii) 图 3.22 的成本矩阵运用上面的方法.

9. 寻找一个 3×3 的成本矩阵, 对于这个矩阵, 通过以不同的顺序对行和列做减法, 我们得到不同的初始下界 (有着不同的 0 项).

10. 构建得到下面各结果的 5×5 矩阵的近似行程构造:

(a) 最优行程; **(b)** 成本相当高的行程 (至少超过真实最小值的 50%).

3.4 排序算法的树分析

计算机科学中的一个基本组合过程是对项集合进行排序. 根据项自身的数值或者某个相关变量的值对这些项排序. 为了简化我们的讨论, 假设所有项都有不同的数值.

我们提到的若干排序过程显式地使用了二叉树. 所有过程都隐式地使用二叉检测树, 即, 这些过程使用比较各项对的检测序列. 排序的二叉检测树模型有一个重要的结论. 如果有 n 个项, 则这棵树必须至少有 $n!$ 个叶子, 因为这个集合有 $n!$ 种可能置换, 或重新排列. 根据 3.1 节的定理 4, 二叉检测树的高度一定至少等于 $\lceil\log_2 n!\rceil$, 这个高度近似于 $n\log_2 n$. 因此得到下面的定理.

定理 *在最坏的情况下, 排序 n 个项所需的二分比较的次数至少是* $O(n\log_2 n)$.

这个定理涉及最坏情况, 因为某些排序算法可以非常快地排序某个集合 (对应于二叉检测树中的短路径). 还可以证明排序 n 个项所需的二分比较的平均次数的最好阶是 $O(n\log_2 n)$(见练习 3). 函数 $g(n)$ 是阶 $O(f(n))$ 的, 如果存在常量 c, 使得对于充分大的 n 都有 $g(n)\leqslant cf(n)$.

最著名的排序算法称为**冒泡排序**(bubble sort), 之所以这样命名, 是因为较小的项在列表中上升的样子如同液体中气泡上升一样. 这个算法可以简洁地写成:

```
FOR m ⟵ 2 TO n DO
      FOR i ⟵ n STEP −1 TO m DO
            IF A_i ⩽ A_i-1, THEN interchange items A_i and A_i-1
```

这一过程总是需要 $(n-1)+(n-2)+\cdots+1=1/2n(n-1)$ 次二分比较. 因此, 相对于 $O(n\log_2 n)$ 次比较的理论界限, 这一过程需要 $O(n^2)$ 次比较. 为了清楚 $O(n^2)$ 比 $O(n\log_2 n)$ 的增长快到什么程度, 观察到对于 $n=50$, 有 $n^2=2\,500$ 而 $n\log_2 n\approx 300$, 对于 $n=500$, $n^2=250\,000$, 而 $n\log_2 n\approx 4\,500$.

能够达到 $O(n\log_2 n)$ 界的最好陈述的排序程序是**归并排序**(merge sort). 这一排序过程递归地把原始列表以及随后的子列表一分为二 (或尽可能接近一半), 直到每一个子列表由一个项组成. 然后, 它再按顺序归并这些子列表. 划分过程自然地被表示成平衡二叉树, 而归并过程被表示成为这棵树的反射像.

例 1 归并排序

使用归并排序, 排序列表 5, 4, 0, 9, 2, 6, 7, 1, 3, 8.

划分树和随后的有序归并如图 3.24 所示. □

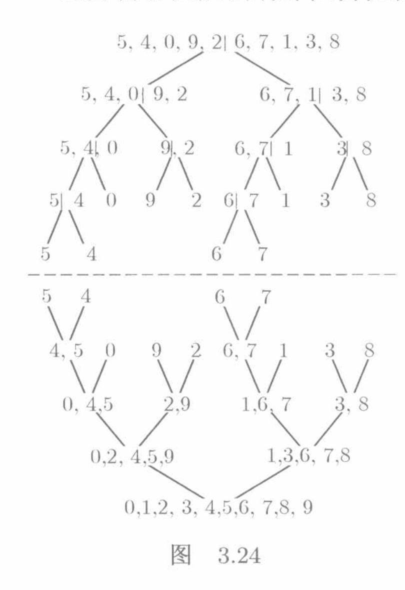


图 3.24

为了分析在归并排序中二分比较的次数, 我们简单化地假设 $n=2^r$, 其中 r 是一个整数. 这时, 划分树在层次 1 的顶点上有大小为 2^{r-1} 的子列表. 一般地, 在层次 k 处, 子列表中有 2^{r-k} 个项. 在层次 r 处, 每一个叶子有一个项.

在归并树中, 首先, 叶子对被排序到层次 $r-1$ 的每一个顶点处, 这需要一次二分比较 (查看哪个叶子项在前面). 参见图 3.24. 一般地, 在层次 k 上的每一个顶点处, 我们把它的 2 个孩子处的有序子列表归并成一个有 2^{r-k} 个项的有序子列表, 这一次归并需要 $2^{r-k}-1$ 次二分比较 (这个数值的验证留作练习). 最后, 层次 1 处的两个有 2^{r-1} 个项的有序子列表在根处被归并. 因为层次 k 上有 2^k 个不同顶点, 所以在层次 k 上的所有顶点处二分比较的次数是 $2^k(2^{r-k}-1)$. 对所有层次求和, 我们计算二分比较的总数是:

$$\sum_{k=0}^{r-1}2^k(2^{r-k}-1)=\sum_{k=0}^{r-1}(2^r-2^k)=\sum_{k=0}^{r-1}2^r-\sum_{k=0}^{r-1}2^k$$
$$=2^r r-(2^r-1)=(\log_2 n)n-(n-1)$$

因为 $n=2^r$, 所以 $r=\log_2 n$. 因此, 归并排序中的二分比较次数是 $O(n\log_2 n)$. 然而, 实现初始的划分过程需要额外的计算机时间. 这份额外的工作也只需要 $O(n\log_2 n)$ 个步骤.

另一个基于树且可以达到界 $O(n\log_2 \mathrm{n})$ 的排序过程称为**堆排序**(heap sort). 一个**堆**(heap) 是二分树或几乎二分树 (有些内部顶点可能只有一个孩子), 它的每一个内部顶点的数值大于它的孩子的数值. 图 3.25a 给出了包含数字0到数字9的堆.

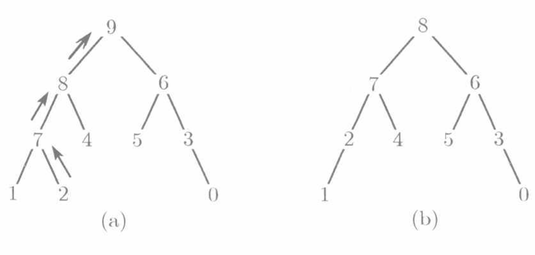


图 3.25

堆的根在集合中有最大值. 如果删除这个堆中的根 (并把它放置于我们正在构建的有序列表的末端), 那么我们可以重新构建堆, 其方法是让被删除的根的较大孩子为新的根, 并且递归地把每一个内部顶点上移, 即用这个顶点的两个孩子中的较大者来替换它的位置. 图 3.25a 中的箭头说明了根 9 被删除时顶点是如何上移的. 图 3.25b 给出新堆. 现在, 我们再一次删除这个堆的根 (并让它成为有序列表中的倒数第二项). 重复这一过程, 直到这个堆被清空, 形成一个有序列表. 这里省略了创建初始堆的重要步骤. 这一问题及堆排序的进一步讨论留作练习.

练 习 3.4

练习概况 前面的练习基于本节给出的排序方法. 余下的练习给出其他排序方案.

1. 运用归并排序, 对下面的序列排序:

(a) 6, 9, 5, 0, 3, 1, 8, 4, 2, 7;

(b) 15, 27, 4, −7, 9, 13, 8, 28, 12, 20, −80.

2. 证明: 把两个有序子列表归并成一个有 n 个项的单一有序列表至多需要 $n-1$ 次比较.

3. 使用 3.1 节练习 17 的结果, 证明排序 n 个项所需要的二分比较次数至少是 $\mathrm{O}(n\log_2 n)$.

4. **(a)** 描述如何从有 n 个项的无序列表构建初始堆 (初始堆应该是平衡树).

(b) 使用问题 (a) 的方法构建练习 1 中各列表的堆.

(c) 对问题 (b) 的堆运用堆排序.

5. 证明: 堆排序需要 $\mathrm{O}(n\log_2 n)$ 比较来排序 n 个项 (其中包括堆的初始构建).

6. 修改下面的排序方法, 使得允许有重复项 (两个或更多相等的项):

(a) 冒泡排序; **(b)** 归并排序; **(c)** 堆排序.

7. 编写下面排序的计算机实现 (使用如 PASCAL 这样的递归语言):

(a) 归并排序; **(b)** 堆排序.

8. 考虑下面称为*树排序*(tree sort) 的排序方案. 我们递归地构建一个 "词典查找" 二叉树. 被排序列表的第一项是根. 根据列表中第二项在数值上是在这个根的前面还是在这个根的后面来确定它是这个根的左孩子还是右孩子. 依次加入每一个后继项作为这个成长起来的树的叶子. 例如, 如果列表的开头几项是 8, 4, 11, 6, $\cdots$, 这 4 个项之后的树如右图所示. 当所有项被结合到一起形成树时, 通过对这个棵树的中序遍历 (3.2 节定义了中序遍历), 我们得到有序序列. 构建练习 1(a) 的列表的树.

8
4 11
6

9. 对于列表 $A_1, A_2, \cdots, A_n (n = 2^r)$, 考虑下面的排序方案. 首先排序 A_i 和 $A_{i+n/2}$(一次比较); 称其为第 i 个有序对, 其中 $i = 1, 2, \cdots, n/2$. 接下来做第 i 有序对和第 $(i + n/4)$ 个有序对的归并排序; 称其为第 i 个 4 元组. 接下来做第 i 个 4 元组和第 $(i + n/8)$ 个 4 元组的归并排序. 持续下去, 直到获得所有 n 个项的有序列表为止.

(a) 对于列表 10, 12, 7, 0, 5, 8, 11, 15, 1, 6, 3, 9, 13, 4, 2, 14, 运用这一排序方法.

(b) 用这一排序方法排序 n 个数的列表需要多少次比较?

3.5 小结及参考文献

本章探究了各种搜索和数据组织问题. 所有这些问题的共同图论工具是树. 3.1 节给出了树的一些基本性质和术语. 3.2 节介绍了生成树并展示了深度优先和广度优先搜索的用途. 3.3 节对一个著名的优化问题, 即旅行商问题, 给出了基于树的分支定界方法和启发式方法. 3.4 节探讨了排序算法背后的决策树.

1847 年, 基尔霍夫发表了一篇关于电子网络的奠基性论文, 这是隐式地使用树的第一篇论文. Cayley 是使用树这一术语的第一人, 1857 年他在关于计数有序树的公式 (3.1 节的定理 5) 中使用了这一术语. 搜索方法已有多年的历史 (参见 Lucas[3]), 但是只有在近年, 随着数字计算机的出现, 这一课题才得到系统发展. 关于搜索、排序和图算法, 有许多优秀的计算机科学图书. 参见 Kruse[1] 或者 Sedgewick[4]. 关于旅行商问题的更好概览, 可以参见文献 [2].

1. R. Kruse, *Data Structures and Program Design in C++*, Prentice-Hall, Upper Saddle River, NJ, 1998.

2. E. Lawler, J. Lenstra, A. Rinnooy Kan, and D. Shmoys, *The Traveling Salesman Problem,* John Wiley, New York, 1985.

3. E. Lucas, *Recreations Mathematiques*, Gauthier-Villars, Paris, 1891.

4. R. Sedgewick, *Algorithms in C*, 3rd. ed. Addison-Wesley, Reading MA, 1997.

第 4 章　网络算法

4.1　最短路径

本章给出若干算法, 并用这些算法解决 4 个重要的网络优化问题: 最短路径, 最小生成树, 最大流及运输问题. **网络**(network) 指的是每条边 e 都配有正整数 $k(e)$ 的图. 这个正整数通常表示一条边的 "长度" 或 "成本", 以公里、美元等为单位, 或表示一条边的容量, 以百万瓦特或加仑每秒等为单位. 我们要讨论的优化问题以各种形式出现于各种管理科学和系统分析环境中. 因此, 对这些问题求解的优良系统程序至关重要. 对于网络流的情况, 我们将看到流优化算法还可以用于证明若干著名的组合定理.

我们从相对简单问题的算法开始, 在网络中寻找从点 a 到点 z 的一条最短路径. 我们不说这条最短路径, 因为一般来说, 存在多条从 a 到 z 的最短路径. 本节的其余部分, 我们都假设所有网络是*无向且连通的*.

我们现在要排除一个可能的最短路径算法: 确定从 a 到 z 的所有路径的长度, 并选出其中一个最短的路径. 计算机很快, 但是它不能那么快: 对于有 20 个顶点的大多数网络, 这样的枚举已经是不可行的. 所以, 当我们发现一个最短路径时, 必须能够证明它是最短的, 而不用明确地把它与其他所有 $a-z$ 路径比较. 尽管这一问题现在听起来很难, 但仍然有一个直接的求解算法.

我们给出的算法归功于 Dijkstra. 这个算法给出了从给定顶点 a 到其他所有顶点的最短路径. 设 $k(e)$ 表示边 e 的长度. 设变量 m 是一个 "距离计数器". 随着 m 的递增, 算法标记到顶点 a 最短矩离为 m 的顶点. 顶点 x 的第一个标签是从 a 到 x 的最短路径上的前一个顶点. x 的第二个标签是从 a 到 z 的最短路径的长度.

最短路径算法

(1) 设 $m=1$, 用 (—, 0) 标记顶点 a("—" 代表一个空格).

(2) 检查每一条从已标记顶点 p 到未标记顶点 q 的边 $e=(p,q)$. 假设 p 的标签是 $[r,d(p)]$. 如果 $d(p)+k(e)=m$, 则用 (p,m) 标记 q.

(3) 如果还有未标记顶点, 那么把 m 递增 1, 并转到步骤 (2). 否则转到步骤 (4). 如果仅对到 z 的最短路径感兴趣, 那么当 z 被标记时, 转到步骤 4.

(4) 对于任意顶点 y, 从 a 到 y 的最短路径的长度是 $d(y)$, 即 y 的第二个标签. 可以如下 (使用第一个标签) 从 y 开始返回而找到这样的一条路径.

我们发现, 这个算法不是专注于到特定顶点的距离, 而是解决这样的问题: 在

1 个单位内, 我们能够到达多远, 在 2 个单位内我们能够到达多远, 3 个单位, ……, m 个单位呢? 我们需要用数学归纳法给出这个算法的形式证明 (对已标记的顶点数进行归纳). 可以证明, 构成从顶点 a 到其他顶点的最短路径的边形成一棵生成树 (练习 11).

这一算法的主要思想是, 为了寻找从 a 到其他任意顶点的最短路径, 我们必须首先寻找从 a 到 "中间" 顶点的最短路径. 如果 $P_k = (s_1, s_2, \cdots, s_k)$ 是从 $s_1 = a$ 到 s_k 的最短路径, 那么 $P_k = P_{k-1} + (s_{k-1}, s_k)$, 其中 $P_{k-1} = (s_1, s_2, \cdots, s_{k-1})$ 是到 s_{k-1} 的最短路径. 类似地, $P_{k-1} = P_{k-2} + (s_{k-2}, s_{k-1})$, 以此类推.

为了记录到 s_k 的一条最短路径, 我们需要存储 (上面算法中标签的第一部分) 的就是 P_k 上倒数第二个顶点, 即 s_{k-1}. 从 a 开始的这条最短路径上 s_{k-1} 前面的顶点是 s_{k-2}, 它是 P_{k-1} 上倒数第二个顶点. 持续这一回溯过程, 我们能够恢复整个 P_k.

上面给出的算法有两个明显低效的地方. 第一, 如果步骤 (2) 中所有和 $d(p) + k(e)$ 都至少有值 $m' > m$, 那么距离计数器 m 应该马上递增到 m'. 第二, 我们无需在每一次 m 增加时重复计算步骤 (2) 中的表达式 $d(p) + k(e)$. 取而代之的是, 给每一个未标记顶点 q 一个 "临时" 标签 $(p, d^*(q))$, 其中 $d^*(q)$ 为: 对于有到 q 的边 $e = (p, q)$ 的已标记顶点 p, $d^*(q)$ 等于 $d(p) + k(e)$ 的最小值. 因此, $d^*(q)$ 代表使用已标记顶点的从 a 到 q 的当前最短路径长度. 现在, 只在新的已标记顶点 p 与 q 相邻时, 才需要更新 q 的临时标签. 上面的内容以及对这里所给 Dijkstra 算法形式的其他改进等详细内容可以在 Ahuja 等人 [1] 中找到.

例 1 最短路径

一对新婚夫妇想要寻找如图 4.1 所示的道路网络中从点 N(尼亚加拉大瀑布) 到点 R(里诺) 的最短路径. 我们运用上面的最短路径算法. 首先, 用 (—, 0) 标记 N. 对于 $m = 1$, 不能做新标记 [我们检查边 (N, b), (N, d), (N, f)]. 对于 $m = 2$, $d(N) + k(N, b) = 0 + 2 = 2$, 我们用 $(N, 2)$ 标记 b; 对于 $m = 3$ 和 4 时, 不能做新标记; 对于 $m = 5$, $d(b) + k(b, c) = 2 + 3 = 5$. 所以用 $(b, 5)$ 标记 c. 继续下去得到图 4.1 所示的标记. 从 R 开始返回, 我们发现最短路径是 $N-b-c-d-h-k-j-m-R$, 且长度等于 24. □

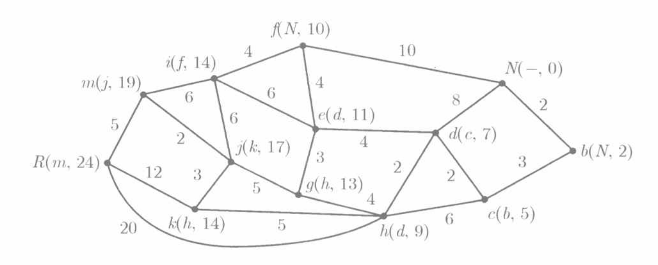


图 4.1

如果要同时寻找所有顶点对之间的最短距离 (不直接寻找所有相关的最短路径), 可以使用下面由 Floyd 开发的简单算法. 设矩阵 D 有项 $d_{ij}=\infty$(或是一个非常大的数), 如果从第 i 个顶点到第 j 个顶点没有边; 否则 d_{ij} 是从 x_i 到 x_j 的边的长度. 于是, 用下面看似简单的计算程序可以给出 Floyd 算法的最易懂描述:

```
FOR k ⟵ 1 TO n DO
    FOR i ⟵ 1 TO n DO
        FOR j ⟵ 1 TO n DO
            IF d_ik + d_kj < d_ij THEN d_ij ⟵ d_ik + d_kj;
```

当程序完成时, d_{ij} 是从第 i 个顶点到第 j 个顶点的最短距离.

练 习 4.1

练习概况 前 6 道练习是关于最短路径的计算. 其余练习讨论相关的理论 (相当容易的理论), 最后一道练习要求编写一个程序.

1. 使用最短路径算法, 寻找图 4.1 中顶点 c 和顶点 m 之间的最短路径.

2. 寻找 4.2 节图 4.2 的网络中下面各顶点对之间的最短路径:

(a) a 和 y; **(b)** d 和 r; **(c)** e 和 g.

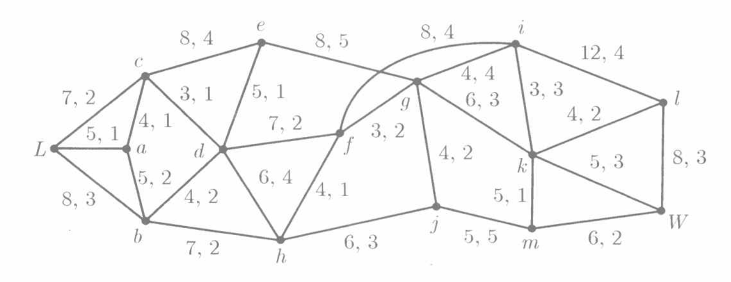
3. 下面的网络给出从 L(小木屋) 成功到达 W(白宫) 的路径. 各边上第一个数是经过这条边所需的时间 (年数), 第二个数是取这条边时所要树立的敌人数量. 使用最短路径算法回答下面的问题:

(a) 寻找最快的成功路径 (从 L 到 W).

(b) 寻找回避点 c, g, k 的最快成功路径.

(c) 寻找从 L 到 W 且树敌数目最少的路径.

(d) 改变最短路径算法的步骤 (2) 中检查边的顺序, 得到另一个树敌最少的路径. 有多少这样的树敌最少的路径?



4. 参考练习 3, 设一条成功的路径的难度指数 R 是 $R=T+2E$, 其中 T 是从 L 到达 W 的时间, E 是树敌总数.

(a) 寻找最通畅 (难度最小) 的成功路径.

(b) 寻找包含边 (f, i) 的最通畅成功路径, 可以在任意方向上经过这条边.

5. 忽略练习 3 中边上的数值, 并使用最短路径算法寻找下面的最短 (最少边) 路径:
   (a) 从 L 到达 W 的最短路径;
   (b) 从 L 到达 W 且包含顶点 d 的最短路径;
   (c) 从 L 到达 W 且包含顶点 d 和顶点 m 的最短路径.
6. 假设练习 3 中的边按着端点字母的顺序给以方向, 其中 L 超前 a, 而 W 跟随在 m 之后 [如果在字母表中 r 超前于 s, 则表示 $(\vec{r}, s)$ 从 r 到 s]. 在这个有向网络中寻找从 L 到 W 的最快路径.
7. 对 m 施用归纳法, 证明 Dijkstra 最短路径算法能够找到网络中从 a 到其他每一个顶点的最短路径.
8. 证明: Floyd 算法能够找到所有顶点对之间的最短路径.
9. 举例说明: 如果允许负边长度, 那么 Dijkstra 算法不真.
10. (a) 证明: 如果适当排列边 [且步骤 (2) 按所给顺序检查边], 那么当存在多条从 a 到 z 的最短路径时, 最短路径算法将产生任意给定的最短路径.
    (b) 排列练习 3 中的图的边, 使得在练习 3(c) 中, 找到的最短路径是 $(L, a, c, d, f, g, k, m, W)$.
11. 证明: 对于最短路径算法, 步骤 (2) 中用于标记新顶点的边形成一棵生成树.
12. 有向图 G 的*传递闭包*(transitive closure) 是这样得到的: 对每一对不相邻顶点 x_i, x_j, 如果从 x_i 到 x_j 有有向路径, 则把边 $(\vec{x_i}, x_j)$ 加到 G 中. 如果 (x_i, x_j) 是 G 中的边, 则设 $d_{ij}=1$, 否则 $d_{ij}=0$. 用下面的 IF 语句取代 Floyd 算法中的 IF 语句

$$\texttt{IF}\quad d_{ik}\cdot d_{kj} > d_{ij}\ \texttt{THEN}\ d_{ij} \longleftarrow 1$$

    证明: 改进的 Floyd 算法能够找到 G 的闭包.
13. 编写实现 Dijkstra 最短路径算法的计算机程序.

4.2 最小生成树

网络的**最小生成树**(minimum spanning tree) 是各边长度 $k(e)$ 之和尽可能小的生成树. 最小生成树存在于各种重要的商业环境中, 诸如寻找连接一组给定场所的最小长度光纤网络. 这一问题看起来比寻找给定两个顶点之间的最短路径要困难些, 但是使用适当的算法, 实际上这一问题要比最短路径问题容易解决. 能够找到简单解决方案的原因是存在一个 "贪婪" 的直接算法, 这一算法通过依次挑选出最短可用边来构建最小生成树.

我们给出两个构建最小生成树的贪婪算法. 设 n 表示网络中顶点的数量.

Kruskal 算法 重复下面的步骤, 直到集合 T 有 $n-1$ 条边 (最初 T 是空的): 向 T 中加入与已经在 T 中的边不形成回路的最短边.

Prim 算法 重复下面的步骤, 直到树 T 有 $n-1$ 条边: 向 T 中加入 T 中的顶点与不在 T 中的顶点之间的最短边 (开始时, 选择任意一条长度最小的边).

对于上面的两个算法, 当要加入的最短边的长度一样时, 我们可以选择其中任

意一条边. 练习 12 给出 Kruskal 算法的正确性证明. 后面给出 Prim 算法的正确性证明.

例 1 最小生成树

寻找图 4.2 的网络的最小生成树. 两个算法都从一条最短边开始. 长度为 1 的边有 3 条: (a, f),(l, q),(r, w). 假设选择边 (a, f). 如果遵循 Prim 算法, 那么我们要加入的下一条边是长度为 2 的边 (a, b), 然后是长度为 4 的边 (f, g), 然后是 (l, q), 然后是 (l, m), 一直下去. 倒数第二条加入的边是长度为 5 的 (m, n) 或 (o, t)[假设选择 (m, n)], 无论加入哪条边, 随后的边都是 (n, o). 最终的生成树由图 4.2 中的黑线表示.

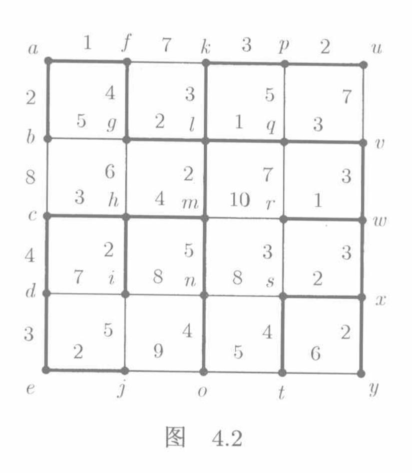


图 4.2

另一方面, 如果遵循 Kruskal 算法, 我们首先包括所有长度为 1 的边: (a, f), (l, q), (r, w). 接下来, 加入所有长度为 2 的边: (a, b), (e, j), (g, l), (h, i), (l, m), (p, u), (s, x), (x, y). 接下来, 加入几乎所有长度 3 的边: (c, h), (d, e), (k, l), (k, p), (q, v), (r, s), (v, w), 但是边 (w, x) 除外, 除非删除 (r, s)[如果这两条边都加入, 则我们将得到一个包含边 (r, w) 和 (s, x) 的回路]. 接下来, 加入所有长度为 4 的边, 最后加入 (m, n) 或 (o, t), 得到与由 Prim 算法产生的最小生成树相同的最小生成树. 这种类似性不是偶然的 (参见练习 14). □

最小生成树问题的难点是, 证明这两个算法生成的树是最小生成树.

定理 Prim 算法生成最小生成树.

证明 为简便起见, 假设所有边都有不同长度. 设 T' 是与由 Prim 算法构造出的树 T^* 有尽可能多共公边的最小生成树.

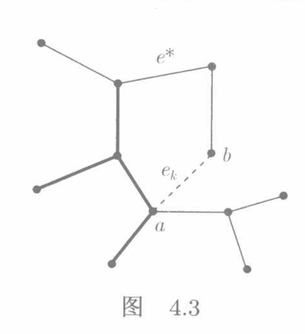


图 4.3

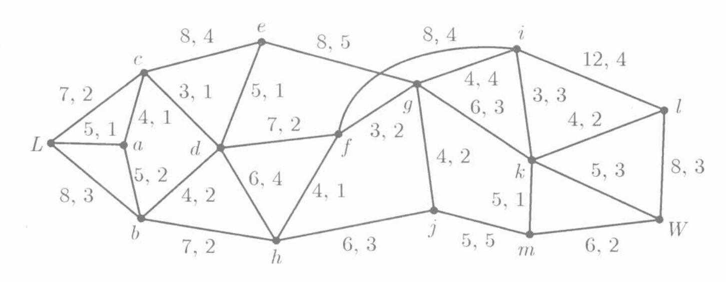
假设 $T^* \neq T'$, 设 $e_k = (a,b)$ 是由 Prim 算法选择的第一条不在 T' 中的边. 这表明由 Prim 算法选择出来的前 $k-1$ 条边组成的子树 T_{k-1} 是最小生成树 T' 的一部分. 在图 4.3 中, T_{k-1} 的边用粗体表示, e_k 的边用虚线表示, 而 T' 的其他边用普通实线表示. 因为 e_k 不在最小生成树 T' 中, 所以 T' 中存在一条连通 a 和 b 的路径 P. P 至少有一条边不在 T_{k-1} 中, 因为否则的话 $P \cup e_k$ 在 T_k 中形成一条回路, 这条回路是由 Prim 算法中前 k 条边形成的. 设 e^* 是 P 中不在 T_{k-1} (参见图 4.3) 中的第一条边 (从 a 开始). 注意, e^* 的一个端顶点在 T_{k-1} 中.

如果 e^* 比 e_k 短, 那么在第 k 次迭代后, Prim 算法将选择 e^* 而不是 e_k. 如果 e^* 比 e_k 大, 我们从 T' 中删除 e^*, 并用 e_k 取代它. 于是新的 T' 仍是生成树 (细节参见练习 10), 但是它的长度更短, 这与原始 T' 的最小性矛盾. ◇

练 习 4.2

练习概况 前 5 道练习构建最小生成树. 练习 6~15 是两个最小生成树算法的变形及理论问题.

1. 下面的网络给出从 L(小木屋) 到 W(白宫) 的成功路径. 边上的第一个数是沿着这条边构建一段高速公路的成本, 第二个数是必须砍倒的树木的数量 (以千计).



(a) 使用 Kruskal 算法, 寻找连通所有顶点的最小成本高速公路集合.

(b) 环保人士提出的法庭诉讼撤销了边 (c, e), (d, f), (k, W) 的使用. 现在, 寻找最小成本高速公路集合.

(c) 寻找两个不相邻顶点, 使得问题 (a) 中的树不包含这两个顶点之间的最廉价路径.

(d) 使用 Prim 算法, 寻找使砍伐树木的数量最少的高速公路集合.

2. 参考练习 1, 假设高速公路集合必须包含城市 c 或 d(不必二者都包含), 以及所有其他城市. 寻找最小成本高速公路集合.

3. 使用下面各算法, 寻找图 4.1 的网络的最小生成树:

(a) Prim 算法; **(b)** Kruskal 算法.

4. 参考练习 1, 假设官员避暑地在边 (f, i) 上. 寻找最小成本高速公路集合, 使得 (f, i) 在这个集合中.

5. 对于图 4.2 的网络, 寻找**最大**生成树 (生成树的边长之和最大).

6. 如果每一条边有不同的成本, 证明最小生成树是唯一的.

7. 修改 Kruskal 算法, 使得它可以寻找包含指定边的最小生成树. 证明你的修改是可行的.

8. 修改 Prim 算法, 得到寻找最大生成树的算法.

9. 在定理的证明中, 证明:

(a) $e_k = (a, b)$ 有一个端顶点 (a 或者 b) 在 T_{k-1} 中;

(b) e^* 的端顶点之一在 T_{k-1} 中.

10. 证明: 由 e_k 取代 e^* 得到的 (定理证明的最后陈述中提到的) 新 T' 是树, 即, T' 连通且没有回路.

11. 设 T 是在无向连通网络 N 中由 Prim 算法找到的生成树.

(a) 证明: 除形成回路的边之外, T 包含 N 中所有最短长度的边.

(b) 证明: 如果 $e^* = (a, b)$ 是 N 的不在 T 中的任意边, 且如果 P 是 T 中从 a 到 b 的唯一路径, 那么对于 P 中每一条边 e, 有 $k(e) \leqslant k(e^*)$.

(c) 证明：问题 (b) 构画出一棵最小生成树, 即, 任意生成树 T(不仅仅是由 Prim 算法生成的树) 是最小生成树, 当且仅当问题 (b) 对 T 总为真.

12. 证明：Kruskal 算法给出一棵最小生成树.

13. 构建有 8 个顶点和 15 条边的无向连通网络, 它有包含每一对顶点间最短路径的最小生成树.

14. 假设我们对无向连通网络 N 的边进行排序, 且对于 Prim 算法和 Kruskal 算法, 当加入的下一条边长度相同时, 选择下标较小的边.

(a) 证明：能够排序边, 使得 Prim 算法产生任意给定的最小生成树.

(b) 证明：能够排序边, 使得 Kruskal 算法产生任意给定的最小生成树.

(c) 证明：对于有序的边, 两个算法给出相同的树.

15. 给定有 n 个顶点的无向连通图 G, 且每条边有指定的长度, 我们构建一个图 G_N, 这个图的顶点对应于 G 的最小生成树, 如果两个顶点 v_1, v_2 所对应的最小生成树 T_1 和 T_2 相差一条边, 即 $T_1 = T_2 - e' + e''$(对于某个 e', e''), 则它们相邻.

(a) 生成有 8 个顶点的网络 G, 使得 G_N 是一个无弦 4 回路.

(b) 证明：如果 T_1 和 T_2 是有 k 条不同边的最小生成树, 即 $|T_1 \cap T_2| = n - k$, 那么在 G_N 中对应顶点间存在长度为 k 的路径.

16. 编写 (尽可能高效地) 实现下面算法的计算机程序：

(a) Kruskal 算法; (b) Prim 算法.

4.3 网 络 流

本节把与网络中的边 e 相关联的整数 $k(e)$ 解释为容量. 我们寻找从顶点 a 到顶点 z 的最大 "流", 使得每条边上的流不超过这条边的容量. 很多运输问题是这种一般形式的特例, 例如, 使流经从休斯顿到纽约的巨大输油管道的油的流量最大 (这里边容量代表每分钟一段管道的油容量, 以桶为单位), 或者使通过从纽约到洛杉矶的电话网络的可能电话呼叫数最大. 为了方便起见, 假设所有网络都是有向的.

我们定义有向网络 N 中的 $a-z$**流**$f(e)$ 为在每一条边 e 上有定义的整数值函数 $f(e)$, $f(e)$ 是 e 中的流, 同时定义满足下面条件的**源**(source) 点 a 和**汇**(sink) 点 z. $\mathrm{In}(x)$ 和 $\mathrm{Out}(x)$ 分别表示顶点 x 的进入边集合和出去边集合.

(a) $0 \leqslant f(e) \leqslant k(e)$.

(b) 对于 $x \neq a$ 或 z, $\sum\limits_{e\in \mathrm{In}(x)} f(e) = \sum\limits_{e\in \mathrm{Out}(x)} f(e)$.

为了简化分析, 我们假设流是从 a 到 z, 在反方向上没有流：

(c) $f(e) = 0$, 如果 $e \in \mathrm{In}(a)$ 或 $e \in \mathrm{Out}(z)$.

图 4.5 给出了一个流的样例, 每条边 e 上的容量和流写在这条边的旁边：$k(e)$, $f(e)$. 整数容量和整数流不会给我们带来制约, 我们的计数单位可以是 $\frac{1}{1\,000}$ 盎司, 而不必是桶. 只有一个供应顶点 (即源点 a) 和一个需求顶点 (即汇点 z) 的需求同样不会给我们带来制约. 其原因是, 可以把有多个源点和多个汇点的网络问题转换

成单源点、单汇点的网络, 如下面的例子所说明的那样.

例 1　有多个供应点和多个需求点的流网络

考虑图 4.4 中粗边表示的带有多个供应点和多个需求点的网络. 顶点 b 的供应总量是 60 个单位流, 而顶点 c 和顶点 d 各自的供应总量是 40 个单位, 顶点 h, i, j 的流需求量分别是 50, 40, 40 个单位. 能够满足这些需求吗? 源点可能是汽油精炼厂, 汇点可能是油车分配中心, 或者源点是工厂, 汇点是仓库.

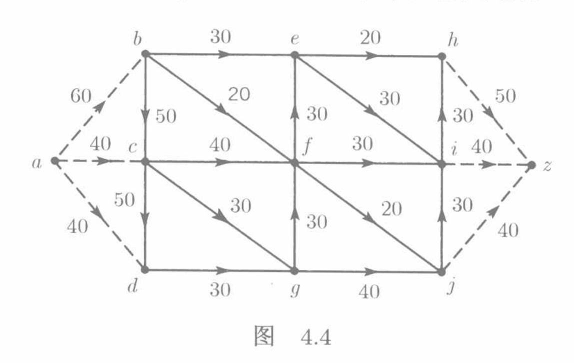


图　4.4

我们用如下的标准单源点单汇点网络来模拟这一网络问题. 使 b, c, d 成为正常的非源点. 加入一个新源点 a, 以及边 $(a^{\rightarrow}, b)$, $(a^{\rightarrow}, c)$ 和 $(a^{\rightarrow}, d)$, 这些边的容量分别为 60, 40, 40. 这一构建模拟限量源点 b, c, d 的角色. 接下来使顶点 h, i, j 成为正常的非汇点, 加入新的汇点 z, 以及边 $(h^{\rightarrow}, z)$, $(i^{\rightarrow}, z)$, $(j^{\rightarrow}, z)$, 这些边的容量分别是 50, 40, 40.

满足原来需求的流等价于新网络中的这样的流: 它能使进入 z 的边 (即, 值为 130 的流) 饱和. 如果不存在容量小于 130 的 $a-z$ 切割, 那么就存在这样的流. □

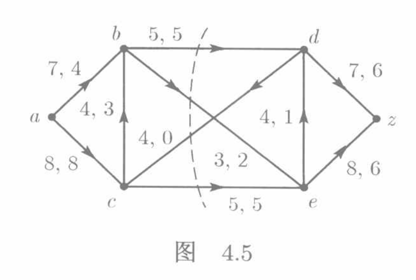


图　4.5

设 $(P, \bar{P})$ 表示所有边 $(x^{\rightarrow}, y)$ 的集合, 其中 $x \in$ 属于 P, $y \in \bar{P}$($\bar{P}$ 是 P 的补). 我们称这样的集合 $(P, \bar{P})$ 为一个**切割**(cut). 图 4.5 中的切割 $(\{a, b, c\}, \{d, e, z\})$ 是由边 $(b^{\rightarrow}, d)$, $(b^{\rightarrow}, e)$, $(c^{\rightarrow}, e)$ 组成的; 边 $(d^{\rightarrow}, c)$ 不在这个切割中, 因为它是从 $\bar{P}$ 到 P 的. 在图 4.5 中, 我们用点线表示这个切割, 它把 $P=\{a,b,c\}$ 中的顶点与 $\bar{P}$ 中的顶点分开; 这个切割中的边是从左到右穿过点线的边. 我们称 $(P, \bar{P})$ 是一个 $a-z$**切割**, 如果 $a \in P$, $z \in \bar{P}$. 网络中的 $a-z$ 切割很重要, 因为从 a 到 z 的所有流都必须穿过每个 $a-z$ 切割. 因此, 任意 $a-z$ 切割中的边的组合容量是从 a 到 z 能获得的流量的上界.

设 $f(e)$ 是网络 N 中的一个 $a-z$ 流, 且设 P 是不包含 a 或 z 的 N 的顶点子

集. 对于每一个 $x \in P$, 对条件 (b) 中的流守恒方程求和, 我们得到

$$\sum_{x\in P}\sum_{e\in \mathrm{In}(x)} f(e)=\sum_{x\in P}\sum_{e\in \mathrm{Out}(x)} f(e)$$

某些 $f(e)$ 项同时在上面等式的两边出现, 即这些流是从 P 中一个顶点到 P 中另一个顶点的边上的 $f(e)$. 从等式两边消去这样的 $f(e)$, 等式左边变成 $\sum\limits_{e\in(\bar{P},P)} f(e)$, 而等式右边变成 $\sum\limits_{e\in(P,\bar{P})} f(e)$. 因此, 我们有:

(b′) 对于每一个不包含 a 或 z 的顶点子集 P,

$$\sum_{e\in(\bar{P},P)} f(e)=\sum_{e\in(P,\bar{P})} f(e)$$

即, 进入 P 的流等于从 P 出来的流. 下面的直观结果可以用 (b′) 验证为真.

定理 1 对于网络 N 中的任意 $a-z$ 流 $f(e)$, 从 a 出来的流等于进入 z 的流.

证明 暂时假设 N 不包含边 $(a^{\rightarrow}, z)$. 设 P 是除 a 和 z 之外的 N 中所有顶点的集合. 所以 $\bar{P}=\{a,z\}$.

记住, 从 $\{a, z\}$ 进入 P 的流必须来自 a, 因为条件 (c) 禁止流从 z 流出. 类似地, 根据条件 (c), 所有从 P 出来的流都必须到 z. 于是, $(P, \bar{P})$ 是由 (从 P 开始) 进入 z 的边组成的, 而 $(\bar{P}, P)$ 是由从 a 出来 (进入 P) 的边组成的. 因此, 有

$$\text{从}a\text{出来的流}=\sum_{e\in(\bar{P},P)} f(e)\overset{\text{由(b′)}}{=}\sum_{e\in(P,\bar{P})} f(e)=\text{进入}z\text{的流}$$

当边 $(a^{\rightarrow}, z)$ 中有流时, 上面的流等式仍然成立. ◇

$a-z$ 流 $f(e)$ 的**值**(value), 记作 $|f|$, 等于从 a 出来的各边上的流的和, 或等价地 (根据定理 1), 等于进入 z 的流的和. 我们考虑 $|f|$ 能够是多大. 它的上界是离开 a 的边的容量之和, 因为

$$|f|=\sum_{e\in \mathrm{Out}(a)} f(e)\leqslant\sum_{e\in \mathrm{Out}(a)} k(e)$$

类似地, 进入 z 的边的容量之和也是 $|f|$ 的上界. 直观上, $|f|$ 不超过切割所有从 a 到 z 的流的任意边集合的容量之和, 即, 不超过 $a-z$ 切割中的边集合的容量之和.

定义切割 $(P, \bar{P})$ 的**容量**(capacity)$k(P, \bar{P})$ 为

$$k(P,\bar{P})=\sum_{e\in(P,\bar{P})} k(e)$$

图 4.5 中的 $a-z$ 切割 $(P, \bar{P})$ 的容量等于 13, 其中 $P=\{a,b,c\}$. 这告诉我们, 在图 4.5 的网络中, 没有值大于 13 的 $a-z$ 流. 这引发下面的定理.

定理 2 对于网络 N 中的任意 $a-z$ 流 f 和任意 $a-z$ 切割 $(P,\bar{P})$, 有 $|f| \leqslant k(P,\bar{P})$.

证明 直观上, $|f|$ 应该等于从 P 到 $\bar{P}$ 的总流量 [它不超过 $k(P,\bar{P})$]. 但是, 不修改网络 N, 我们就无法使用条件 (b′) 来证明这类等式. 我们不能在 N 中使用 (b′), 因为条件 (b′) 要求 P 不包含源点 a.

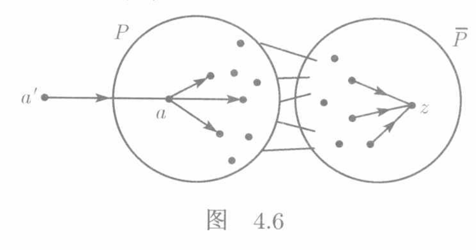


图 4.6

扩展网络 N, 通过向这个网络加入新源点 a', 它有容量巨大的边 $e'=(a'^{\rightarrow},a)$. 指定 e' 的流值等于 $|f|$, 这导致在这个扩展网络中有一个有效的 $a'-z$ 流 (参见图 4.6). 事实上, 原来的源点从 "超源头"a' 得到它自己的流. 注意, a' 是 $\bar{P}$ 的一部分. 观察发现, 新源点 a' 充当类似于我们在例 1 的多源点网络中添加的源点 a 的角色.

在新网络中, 可以把条件 (b′) 运用于 P. (b′) 表明, 至少为 $|f|$ 的进入 P 的流 (其他流可以沿着从 $\bar{P}$ 出来的边进入 P) 等于从 P 出来的流. 因此

$$|f| \leqslant \sum_{e\in(\bar{P},P)} f(e) \overset{\text{由(b')}}{=} \sum_{e\in(P,\bar{P})} f(e) \leqslant \sum_{e\in(P,\bar{P})} k(e) = k(P,\bar{P}) \qquad \Diamond(*)$$

推论 2a 对于网络 N 中的任意 $a-z$ 流 f 和任意 $a-z$ 切割 $(P,\bar{P})$, $|f|=k(P,\bar{P})$, 当且仅当:

(i) 对于每一条边 $e\in(\bar{P},P)$, $f(e)=0$.

(ii) 对于每一条边 $e\in(P,\bar{P})$, $f(e)=k(e)$.

而且, 当 $|f|=k(P,\bar{P})$ 时, f 是最大流且 $(P,\bar{P})$ 是有最小容量的 $a-z$ 流.

证明 考虑前面证明的 $(*)$ 中的两个不等式. 如果条件 (i) 成立, 则第一个不等式变成等式: 从 $\bar{P}$ 出来进入 P 的流 (在扩展网络中) 等于 $|f|$, 否则进入 P 的流大于 $|f|$. 如果条件 (ii) 成立, 则第二个不等式成为等式: 从 P 出来的流等于 $k(P,\bar{P})$, 否则它更小. 因此, $(*)$ 中的等式成立当且仅当 (i) 和 (ii) 都成立. 推论中的最后陈述直接得之于定理 2. $\Diamond$

我们已经详述了最大流算法所需的所有概念. 首先, 我们讨论一项直观但不完善的技术, 有时我们利用这一技术作为取代正确算法的一种捷径. 捷径中的缺点暴露之后, 正确算法就更容易理解和接受.

所有正常流可以分解成从 a 到 z 的**单位流路径**(unit-flow path)(简称 $a-z$ 单位流) 之和 (我们将在练习 23 中讨论不能如此分解的变态流). 例如, 在电话网络中, 从纽约到洛杉矶的电话流可以分解成为各个电话呼叫. 类似地, 在管道网络中的石油流可以分解成为每个单个石油分子的路径. 形式地说, 沿着 $a-z$ 路径 L 的 $a-z$ 单位流 f_L 定义为: 若 e 在 L 中, 则 $f_L(e)=1$; 若 e 不在 L 中, 则 $f_L=0$.

这种分解给出构建最大流的一个方法. 我们通过不断地把 $a-z$ 单位流加到一

起来构建尽可能大的流, 并总保证不超过任意边的容量. 加入的单位流只使用**未饱和**(unsaturated) 边, 即现存的流不等于容量的那些边. 所以必须构建由未饱和边组成的路径. 使用流 f 来定义边 e 的**盈余**(slack)$s(e)$ 为

$$s(e) = k(e) - f(e)$$

如果 s 是 $a-z$ 单位流 f_L 的 $a-z$ 路径各边盈余的最小值, 那么我们沿着 L 注入 $sf_L = f_L + f_L + f_L + \cdots + f_L$($s$ 倍) 的附加流.

如果 $f_1, f_2, \cdots, f_m$ 是 $a-z$ 单位流, 那么 $f = f_1 + f_2 + \cdots + f_m$ 将满足流定义中的 (本节一开始给出的) 条件 (b) 和 (c), 因为 $f_1, f_2, \cdots, f_m$ 满足这些条件. 如果 f 还满足条件 (a), 即, 对于所有的 e, $f(e) \leqslant k(e)$, 那么 f 是一个有效流.

例 2 使用流路径构建流

我们使用刚刚描述的方法构建图 4.7a 的网络的最大 $a-z$ 流. 注意, 这个流不会超过 10, 这是它的一个上界, 也就是, 不超过从 a 出来的边容量之和.

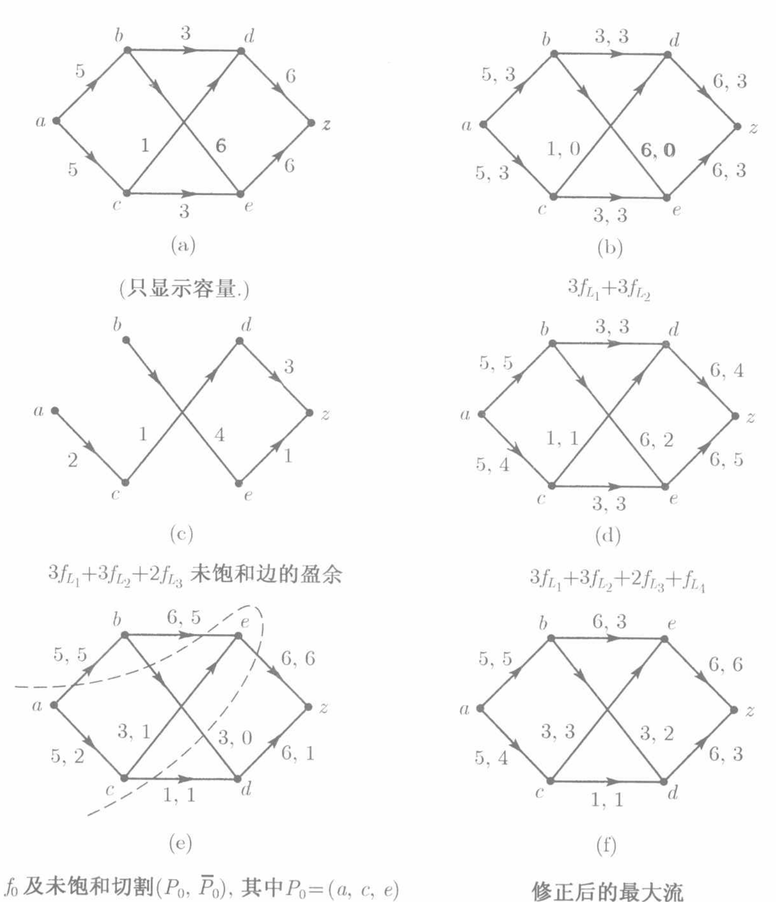


图 4.7

我们从零流开始, 即, 对于所有 e, $f(e)=0$. 现在, 寻找某条从 a 到 z 的路径, 例如, $a-z$ 路径 $L_1=a-b-d-z$. L_1 上的最小盈余是 3(开始时, 每条边的盈余等于它的容量). 所以向我们的初始零流加入 $3f_{L1}$. 除边 $(\vec{b}, d)$ 之外的所有边都仍然是未饱和的.

假设接下来寻找 $a-z$ 路径 $L_2=a-c-e-z$, 它的最小盈余也是 3. 当前流是 $3f_{L1}+3f_{L2}$, 如图 4.7b 中所示. 使用有最小盈余 2 的路径 $L_3=a-b-e-z$ 得到增广流 $2f_{L3}$. 图 4.7c 给出其余未饱和边及它们的盈余.

图 4.7c 中唯一的 $a-z$ 路径是 $L_4=a-c-d-z$, 其最小盈余是 1. 加入流 f_{L4}(参见图 4.7d) 之后, 我们再无法从 a 到 c. 观察发现, 我们在 $a-z$ 切割 $(P, \bar{P})$ 中有饱和边, 其中 $P=\{a, c\}$. 最后的这个流值 9 等于这个切割的容量 $k(P, \bar{P})$, 所以根据推论 2a 知道, 这个流是最大的. □

例 3 不完善流构建

现在, 把图 4.7a 中的网络重新画成如图 4.7e 的样子, 即, 互换 d 和 e 的位置.

再一次选择穿越这个网络的顶部和底部的增广 $a-z$ 流路径, 此时顶部与底部的增广流路径的流值分别是 5 和 1. 因为边 $(\vec{a}, b)$ 和 $(\vec{c}, d)$ 经过这些流路径而饱和, 所以唯一可能的未饱和边 $a-z$ 路径是 $L_5=a-c-e-z$, 有最小盈余 1[这个最小盈余正好出现在边 (e, z) 上]. 加入 $a-z$ 单位流 f_{L5} 之后, 我们得到图 4.7e 所示的流 f_0. 切割 $(P_0, \bar{P}_0)$ 是饱和的, 其中 $P_0=\{a,c,e\}$, 所以不再存在增广 $a-z$ 单位流. 而 $|f_0|=7$, $k(P_0, \bar{P}_0)=12$. 回想一下, 在例 2 中, 对于相同的网络, 我们却得到大小为 9 的流. 发生了什么?

现在, 我们看到任意增广 $a-z$ 单位流序列不一定产生最大流. 我们有这样的一个流 f_0 和一个饱和 $a-z$ 切割 $(P_0, \bar{P}_0)$ 使得 $|f_0|<k(P_0, \bar{P}_0)$. 由推论 2a 可知, 对某条边 $e\in(\bar{P}_0, P_0)$, 一定存在某个流. 观察图 4.7e, 我们看到流路径 $L'=a-b-e-z$ 在边 $(\vec{b}, e)$ 上反方向穿越 $a-z$ 切割 $(P_0, \bar{P}_0)$(从 $\bar{P}_0$ 到 P_0), 或等价地, L' 向前穿越这个切割两次, 第一次在边 $(\vec{a}, b)$ 上, 第二次在边 $(\vec{e}, z)$ 上. 因此, L' 上的 5 个单位的流消耗这个切割中容量的 10 个单位, 因此 $k(P_0, \bar{P}_0)$ 比 $|f_0|$ 大 5 个单位.

本例中的增广流路径序列不产生最大流, 其原因可以直观地解释如下. 因为沿 L' 传送 5 个单位流 (参见图 4.7e), 我们已经经由 b 到达 e 发送了整个流, 而没有剩余单位流到达 d. 因此, 只有经由 c 的 1 个单位流可以发送到 e, 且在边 $(\vec{e}, z)$ 上. 但是经由 c 的这个流的大部分必须到达 e, 因为 $(\vec{c}, d)$ 的容量只有 1. 总之, 最初 $a-b-e-z$ 上的 5 个单位流是一个 "错误", 因为边 $(\vec{e}, z)$ 上的一部分容量为从 c 出来的流保存了起来. □

如何能够避免或改正这样的错误呢? 如果我们已经了解这一错误出现在哪里, 那么就有可能改变其中一些流路径, 并尝试启用新的增广流路径序列. 然而, 在随后的构建中, 我们很有可能犯其他错误. 当然, 也可能存在某些网络, 在这些网络中, 无论怎样使用流路径序列, 我们都可能不犯这样的错误. 用切割来说, 我们最终总

是可以得到一个饱和的 $a-z$ 切割, 而我们的一个流路径穿越它两次.

幸运的是, 有一个改正 "错误" 的过程, 从而使流增大. 这一方法并不是寻找必须为后来的流路径保存的边, 例如像上面的例子所说的那样 (这是一个相当困难的问题). 相反, 它寻找在错误的道路上 (反向) 穿越一个 $a-z$ 切割的流. 然后, 它寻找减少这个反向流但不改变穿越这个切割的正向流的方法, 其结果是整体流的更大部分穿越这个切割 (并经过这个网络). 下面的例子给出了这一过程背后的思想.

例 3(续) 边 $(b^{\rightarrow}, e)$ 含有 5 个单位反向穿越饱和切割 $(P_0, \bar{P}_0)$ 的流, 其中 $P_0 = \{a, c, e\}$. 边 $(b^{\rightarrow}, e)$ 中的流能够减少多少呢?

流的条件 (b), 即, 在每一个顶点处进入流等于出去流, 要求从 b 到 e 的进入流的减少量必须由来自于 P_0 的其他地方到 e 的流的增加而得到补偿 (如果补偿流来自于 $\bar{P}_0$, 那么我们将有一个新的反向流). 这样的增量最终来自 a. 因此, 我们需要 P_0 中的一个 $a-e$ 流路径. 唯一这样的路径是 $K_1 = a-c-e$. 类似地, 从 b 到 e 的出去流的减少也必须由从 b 到 P_0 的其他地方的出去流的增加而得到补偿. 所以我们需要 $\bar{P}_0$ 中的一条 $b-z$ 流路径. 这样的唯一路径是 $K_2 = b-d-z$.

沿 K_1 的最小盈余是 2, 沿 K_2 的最小的盈余是 3. 于是我们可以把 $(b^{\rightarrow}, e)$ 上的流减去 2, 同时把 K_1 和 K_2 中的流增加 2. 图 4.7f 给出最终的最大流. 注意, 这个新的最大流不同于图 4.7d 中的最大流, 尽管二者都使 $a-z$ 切割 $(\{a, c\}, \{b, d, e, z\})$ 饱和. □

有向图中的**链**(chain) 是不计边方向时形成一条路径的边序列. 沿着 $a-z$ 链 K 的从 a 到 z 的**单位流链**(unit-flow chain) 是这样的 "流" f_K: 在 K 的每一条正向边上值为 1, 在 K 的每一条反向边上值为 -1, 其他边上值为 0. 注意, f_K 不是一个真实的流, 因为它可以假设在某些边上的值为负数. 但是, 如果 f 在 K 的每一条反向边已经有正值, 而在 K 的每一条正向边上有盈余, 那么 $f + f_K$ 是有效流.

例 3(续) 中的流修正是由沿链 $K = a-c-e-b-d-z$ 上的 $a-z$ 流链 (2 单位) 组成的. 当在一个流链上没有反向边出现时, 那么这条流链就是流路径. 我们将看到任意增广 $a-z$ 流链序列都可以扩大成最大流. 流链是流路径的适当扩展, 它既能构建额外的流, 同时又能修正可能的 "错误".

现在给出一个增加流值的流链算法. 这个算法是这样设计的: 如果它不能得到一个增广 $a-z$ 流链, 那么它就将产生一个饱和的 $a-z$ 切割, 这个 $a-z$ 切割的容量等于当前流 f 的值 $|f|$. 因此, 根据推论 2a, f 是最大的.

这个算法按 4.1 节最短路径算法的方式递归地尝试构建从 a 到所有顶点的增广流链. 这一算法给顶点 q 分配两个标签: $(p^{\pm}, \Delta(q))$, 其中 p 是从 a 到 q 的流链上的前面顶点, 而 $\Delta(q)$ 是从 a 到 q 传送的额外流量. 即, $\Delta(q)$ 是从 a 到 q 的流链上各边的最小盈余. 在一条反向边 e 上, 这个盈余是可以减去的流量, 即 $f(e)$. p 上的上标 + 表示把流加入到边 $(p^{\rightarrow}, q)$ 上, 而另一个上标 − 表示从边 $(q^{\rightarrow}, p)$ 中减去的流.

增广流算法

(1) 给顶点 a 赋标签为 $(-, \infty)$.

(2) 使用第二个标签 $\Delta(p)$ 调用要扫描的顶点 p. 初始的 p 是 a.

(a) 检查每一条进入边 $e=(q^{\rightarrow}, p)$. 如果 $f(e)>0$ 且 q 没有赋标签, 那么用 $[p^-, \Delta(q)]$ 给顶点 q 赋标签, 其中 $\Delta(q)=\min[\Delta(p), f(e)]$.

(b) 检查每一条出去边 $e=(p^{\rightarrow}, q)$. 如果 $s(e)=k(e)-f(e)>0$, 且 q 没有赋标签, 那么用 $[p^+, \Delta(q)]$ 给 q 赋标签, 其中 $\Delta(q)=\min[\Delta(p), s(e)]$.

(3) 如果 z 已经被赋标签, 转移到步骤 (4). 否则, 选择另一个已赋标签顶点进行扫描 (它不是前面扫描过的顶点), 并转移到步骤 (2). 如果没有要扫描的其他已赋标签顶点, 那么设 P 是已赋标签顶点的集合, 此时, $(P, \bar{P})$ 是一个饱和的 $a-z$ 切割. 另外, $|f|=k(P,\bar{P})$, 因此 f 是最大的.

(4) 如在最短路径算法中那样, 从 z 回溯, 寻找盈余边构成的 $a-z$ 链 K. 然后, 沿 K 上的有 $\Delta(z)$ 个单位流的 $a-z$ 流链 f_K 就是我们希望的增广流. 把 K 的边的流增加 $\Delta(z)$ 个单位 (如果某条边在 K 中是反向边, 则减去流).

如同最短路径算法一样, 上面的流算法把部分流链从当前已赋标签顶点扩展到相邻的未赋标签顶点, 而且用于标签顶点的边形成生成树. 在证明反复使用这一算法总是可以产生一个最大流这一事实之前, 我们再给出些例子.

例 3(续) 对图 4.7e 中的流运用增广流算法.

根据算法的步骤 1, 给顶点 a 赋标签 $(-, \infty)$. 接下来, 我们在顶点 a 处运用步骤 2b(不能在 a 处运用步骤 2a, 因为流的定义不允许有进入源点 a 的流). 边 $(a^{\rightarrow}, b)$ 是饱和的, 但是边 $(a^{\rightarrow}, c)$ 有盈余 $5-2=3$. 所以, 我们给 c 赋标签 $(a^+, 3)$. 在顶点 c 处, 步骤 2a 没有找到来自于未赋标签顶点的进入流, 但是步骤 2b 发现进入未赋标签顶点 e 的边 $(c^{\rightarrow}, e)$ 上有盈余. 我们给 e 赋标签 $(c^+, 2)$[2 是 $\Delta(c)$ 的最小值, 这是我们到达 c 时的额外流, 是边 $(c^{\rightarrow}, e)$ 上的盈余]. 在顶点 e, 步骤 2a 发现进入未赋标签顶点 b 的边 $(b^{\rightarrow}, e)$ 上的正流. 我们给 b 赋标签 $(e^-, 2)$. 从 b 开始, 我们给 d 赋标签 $(b^+, 2)$, 从 d 开始, 我们给 z 赋标签 $(d^+, 2)$. 此时, 汇点 z 已被赋标签, 所以步骤 (4) 告诉我们得到从 a 到 z 的流, 流值增加 $\Delta(z)=2$. 使用这些标签回溯, 流链 K(从后前向的顺序) 是 $z-d-b-e-c-a$. 新流 f_0+2f_K 如图 4.7f 所示. □

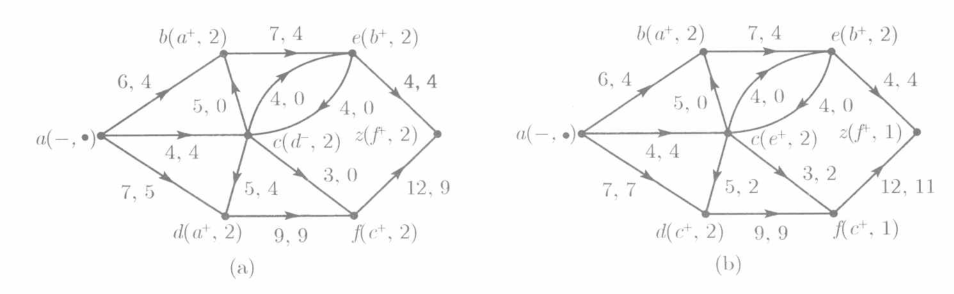
例 4 使用增广流算法

考虑图 4.8a 中的网络. 寻找从 a 到 z 的最大流.

如果由计算机去寻找最大流, 那么它一定从零流开始. 人力解决关于流问题时, 可以通过观察得到一个非零流, 并从这个非零流开始, 从而加速寻找最大流的过程 (在较小的网络中, 通常可以通过观察得到一个最大流). 观察图 4.8, 设最初的流 $f=4f_{K_1}+4f_{K_2}+5f_{K_3}$, 其中, $K_1=a-b-e-z$, $K_2=a-c-d-f-z$,

$K_3 = a - d - f - z$. 读者也许会发现我们从 c 到 d 传送了过量的流. 其实, 这个流的一部分应该直接从 c 发送到 f.

现在, 我们对流 f 运用标签算法. 给顶点 a 赋标签 (—, ∞). 在顶点 a 处扫描边, [根据流定义的部分 (b)], 不可能存在进入源点的带流边, 但是存在 3 条出去边: 边 $(a^{\rightarrow}, b)$ 有盈余 $s(a^{\rightarrow}, b) = 2 > 0$, 且 b 没有被赋标签, 所以我们给 b 赋标签 $(a^+, 2)$, 其中 2 是 $\Delta(a)(=\infty)$ 和 $s(a^{\rightarrow}, b)$ 的最小值; 边 $(a^{\rightarrow}, c)$ 没有盈余; 边 $(a^{\rightarrow}, d)$ 有盈余 $s(a^{\rightarrow}, d) = 2$, 且 d 没有被赋标签, 所以给 d 赋标签 $(a^+, 2)$.



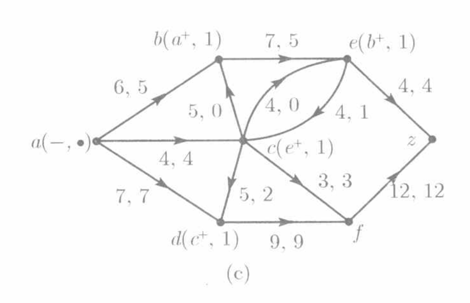


图 4.8

接下来, 扫描 b(我们按顶点被赋标签的顺序扫描顶点). 在顶点 b 有两条进入边: 边 $(a^{\rightarrow}, b)$ 有 $f(a^{\rightarrow}, b) = 4 > 0$, 但是 a 已赋标签; 边 $(c^{\rightarrow}, b)$ 没有流. 在顶点 b 处有一条出去边: 边 $(b^{\rightarrow}, e)$ 有盈余 $s(b^{\rightarrow}, e) = 3$ 且 e 未赋标签, 所以给 e 赋标签 $(b^+, 2)$, 其中 2 是 $\min[\Delta(b), s(b^{\rightarrow}, e)]$.

下面, 我们在 d 处扫描. 在 d 处有两条进入边: 边 $(a^{\rightarrow}, d)$ 来自于已赋标签顶点; 边 $(c^{\rightarrow}, d)$ 有 $f(c^{\rightarrow}, d) = 4$, 而且 c 未赋标签, 所以使用步骤 2a, 我们给 c 赋标签 $(d^-, 2)$, 其中 $2 = \min[\Delta(d), f(c^{\rightarrow}, d)]$. 在 d 处有一条出去边: 边 $(d^{\rightarrow}, f)$ 是饱和的.

从 e 开始不再赋标签. 在 c 处, 我们给 f 赋标签 $(c^+, 2)$. 在 f 处, 我们给 z 赋标签 $(f^+, 2)$.

因为此时 z 已被赋标签, 标签过程终止. 可以在增广 $a - z$ 流链 f_{K4} 上发送 $2[=\Delta(z)]$ 个单位流, 其中 $K_4 = a - d - c - f - z$, 这条链是按回溯过程找到的. 回想一下, 因为边 $(c^{\rightarrow}, d)$ 在 K_4 中是反向边, 所以这条流链从边 $(c^{\rightarrow}, d)$ 中的当前流

中减去 2 个单位. 如图 4.8b 所示, 新流 $f' = f + 2f_{K4}$.

我们消去所有当前标签, 并从头开始标签算法. 给顶点 a 赋标签 (—, ∞). 从 a 开始, 我们给 b 赋标签 (a^+, 2). 在 b 处, 给 e 赋标签 (b^+, 2). 在 e 处, 给 c 赋标签 (e^+, 2). 在 c 处, 给 d 赋标签 (c^+, 2), 给 f 赋标签 (c^+, 1). 在 d 处, 我们不赋新标签. 在 f 处, 给 z 赋标签 (f^+, 1). 增广 $a-z$ 流链是 f_{K5}, 且 $K_5 = a-b-e-c-f-z$ 使用正向边 ($e^{\rightarrow}$, c). 如图 4.8c 所示, 此时的新流是 $f'' = f + 2f_{K4} + f_{K5}$.

读者也许已经发现 f'' 是最大的. 此时 z 处的进入边是饱和的. 让我们对 f'' 再次运用增广流算法. 在顶点 a 处, 我们给 b 赋标签; 在 b 处, 给 e 赋标签; 在 e 处, 给 c 赋标签; 在 c 处, 给 d 赋标签. 再没有其他顶点需要赋标签. 设 P 是已赋标签顶点的集合. 那么 $(P, \bar{P})$ 是由算法描述的饱和 $a-z$ 切割, 且 $|f''| = 16 = k(P, \bar{P})$.

□

在运用这一算法的过程中, 我们必须总是用步骤 2a 检查进入边, 来查看顶点是否有负标签, 尽管这样的负标签频率非常低. 回想一下, 负标签对应一个要修正的错误的流分配. 一种可行的捷径是只使用正标签 (如在前面所讨论的错误过程中那样) 直到没有发现新流路径为止, 然后运用整个算法去搜寻“错误”. 这一捷径的使用要求我们严格地证明: 对任意给定流反复运用算法就可以产生一个最大流.

定理 3 *对于任意给定的 $a-z$ 流 f, 有限次使用增广流算法产生一个最大流. 而且, 如果 P 是最后一次 (不成功地) 运用算法的过程中被赋标签的顶点的集合, 那么 $(P, \bar{P})$ 是一个最小 $a-z$ 切割.*

证明 定理的证明分两个主要部分. 首先, 如果 f 是当前流, 且 f_K 是由这一算法找到的 (沿链 K 的) 增广 $a-z$ 单位流链且 $m = \Delta(z)$, 那么必须证明新流 $f + mf_K$ 的确是网络中的合法流. f 和 f_K 二者都满足流条件 (b) 和 (c) 且取整数值. 因此, $f + mf_K$ 也满足条件 (b) 和 (c), 且取整数值. 标签算法的设计保证 $m = \Delta(z)$ 是流链 K 上的 (适当类型的) 最小盈余, 因此, $f + mf_K$ 满足流条件 (a): $0 \leqslant f(e) + mf_K(e) \leqslant k(e)$.

因为 m 是一个正整数, 所以每一个新流都增加一个整数量. 容量和边数是有限的, 所以算法最终终止: 不给 z 赋标签. 设 P 是算法终止时已赋标签顶点的集合. 显然 $(P, \bar{P})$ 是一个 $a-z$ 切割, 因为 a 已被赋标签, 而 z 没有被赋标签. 观察发现, 不可能存在从一个已赋标签的顶点 p 到一个未赋标签的顶点 q 的不饱和边, 否则在 p 处运用步骤 2a 将给 q 赋标签. 类似地, 在从一个未赋标签的顶点 q 到一个已赋标签的顶点 p 的边上也不可能存在流, 否则在 p 处运用步骤 2a 将给 q 赋标签. 因此, 推论 2a 的两个条件都得到满足. 从而最终的流值等于 $k(P, \bar{P})$, 且是最大的. 而且, $(P, \bar{P})$ 是最小 $a-z$ 切割. ◇

推论 3a(最大流最小切割定理)

在任意有向流网络中, 最大 $a-z$ 流的值等于最小 $a-z$ 切割的容量.

现在, 我们简要说明如何使用有向网络流为有向网络和无向网络中的各种问题

建模. 在例 1 中, 我们已经展示了如何用单一源点和单一汇点的网络为有若干个供应和需求的网络建模.

例 5 无向网络

假设图 4.9 中的无向网络表示一个电话线网络 (一条边的容量是这条电话线所能处理的呼叫数量). 我们希望知道这个网络在两个位置 a 和 z 之间能够同时处理的呼叫的最大数量. 即要寻找这个网络的最大流的值.

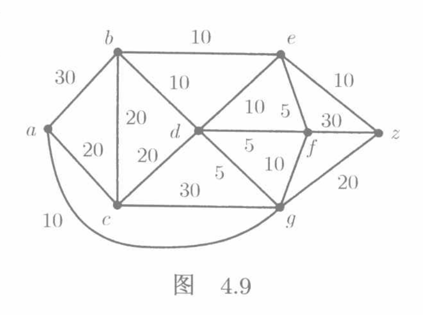


图 4.9

为了使网络有向, 可以用两条边 $(\overrightarrow{x}, y)$ 和 $(\overrightarrow{y}, x)$ 取代每一条无向边 (x, y), 每一条有向边都与无向边 (x, y) 有相同的容量. 一个等价的方法是允许无向边中有有向流. 如果 $e = (x,y)$, 那么 $f(e)$ 是一个带有 "箭头" 的数, "箭头" 表示流是从 x 到 y 还是从 y 到 x. 我们对流算法的步骤 (2) 做如下修改: 当在一个已赋标签的顶点 p 处检查边时, 把带有向内的流的边看成进入边, 而把没有流或者有离开 p 的流的边看成出去边. □

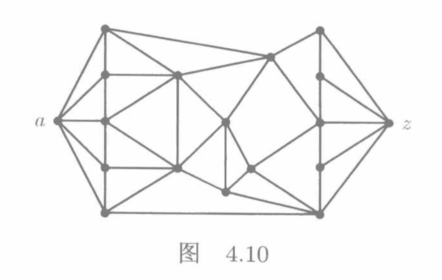


图 4.10

例 6 图中的边不相交路径

我们要在图 4.10 中从 a 到 z 运送信使. 因为某些边 (道路) 可能被封锁, 所以需要每名信使使用不同的边. 可以运送多少名信使? 即, 我们想要知道边不相交路径的数量.

通过给每一条边分配单位容量, 我们把这样的路径问题转化成网络流问题. 可以把流想象成为 "信使流", 而单位容量的意思是每条边最多可以被一名信使使用. 因此, 边不相交路径的数量 (信使的数量) 等于这个无向网络的最大流的值. (对于无向网络中的流, 参见例 5.)

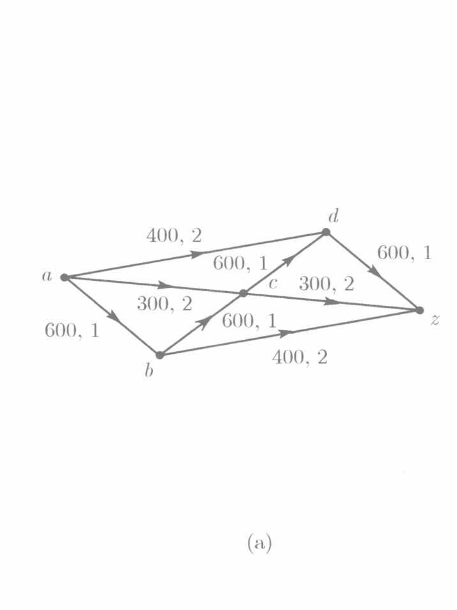
观察发现, 我们已经 (隐式地) 证明了: 单位容量网络的最大 $a-z$ 流问题, 等价于在相关的图中寻找边不相交 $a-z$ 路径的最大数量 (忽略图中边的容量). 通过把每个容量为 k 的边 (x, y) 替换为 k 条多重单位容量边, 并进行上述的转换, 我们可以使用多重图把这一等价性扩展到所有网络. □

例 7 动态网络流

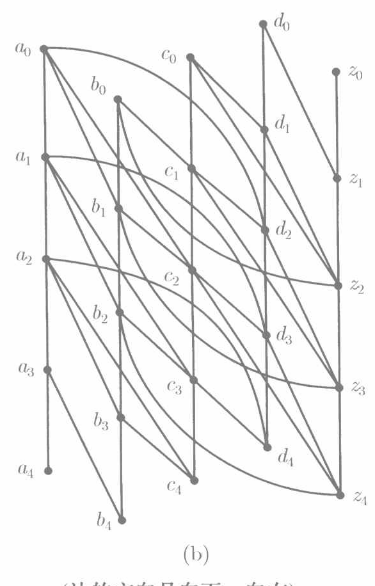
我们想要知道在图 4.11a 的网络中, 4 天之内能把多少辆汽车从位置 a 运送到位置 z. 假设每条边 $(\overrightarrow{x}, y)$ 是这样的路径: 一列火车每天从位置 x 运输到位置 y. 一条边上的第一个数值是这列火车的容量 (汽车的数量). 第二个数字是运输所需的天数. 可以把汽车临时停放在这个网络中的任意位置.

通过给网络加入时间维, 我们把这一动态问题转化成为静态的 $a-z$ 流问题: 每一个顶点 x 被 5 个顶点 x_0, x_1, x_2, x_3, x_4 取代, 其中下标表示 4 天的运输时间内

的第 i 天 (开始的那天是第 0 天). 对于每一条需要 k 天走完的原始边 $(x^{\rightarrow}, y)$, 我们做边 $(x_0^{\rightarrow}, y_k)$, $(x_1^{\rightarrow}, y_{k+1})$, $\cdots$, $(x_{4-k}^{\rightarrow}, y_4)$, 每条边与 $(x^{\rightarrow}, y)$ 有相同的容量. 对于每一个顶点 x, 我们做 4 条形如 $(x_i^{\rightarrow}, x_{i+1})$ 的边, 且有巨大的容量 (事实上, 是无限大). 这些边对应于这样的规定: 允许汽车在任意顶点处临时停留到第二天. 参见图 4.11b.



(a)



(b)

(边的方向是向下、向右)

图 4.11

新网络的最大流给出了原始网络的最大动态流. 注意, 对于除 a 和 z 之外的每一个顶点, 可用顶点的下标范围至多是 1 到 3. 容易把并非每日运输的火车编入这个模型中. □

我们希望上面的例子所描述的基础静态 $a-z$ 流模型的多功能性能给读者留下深刻的印象. 在练习中可以发现更多的例子.

练 习 4.3

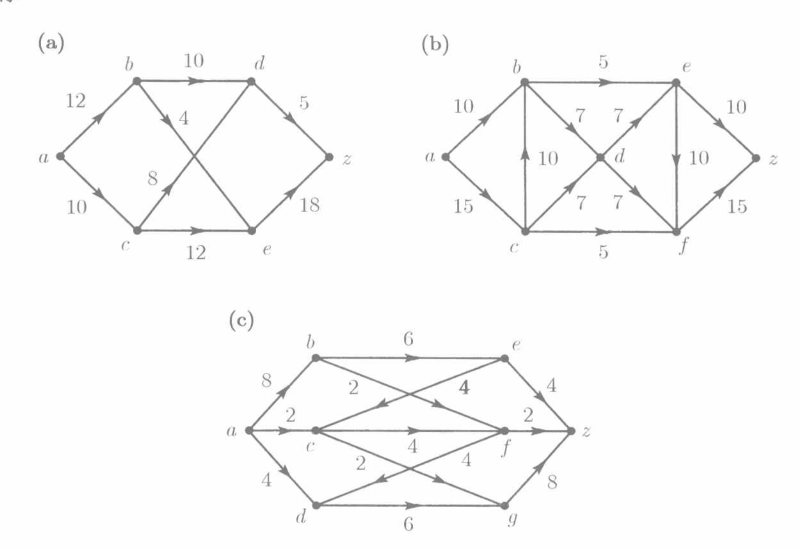
练习概况 练习 1~17 模仿或扩展例 3 至例 7 中的流计算和流模型. 练习 18~38 拓展网络流理论 (后面的练习具有相当的挑战性). 练习 39~42 给出程序设计项目.

在下面的问题中, 除非有特别的提示, 否则读者应该通过观察寻找大部分的流, 然后, 当得到一个近似优化流时, 使用增广算法得到最大流和最小切割.

1. 对图 4.5 中的流运用增广流算法.

2. 在下面的网络中, 寻找最大 $a-z$ 流和最小容量 $a-z$ 切割:

3. 在图 4.9 的网络中 (使用相关的有向网络) 寻找从 a 到 z 的最大流. 并给出最小容量 $a-z$ 切割.



4. 对于 4.1 节练习 3, 把分配给每条边的第一个数值看成容量, 并设边的方向依其端顶点的字母顺序, 且 L 在 a 之前, W 在 n 之后 [例如, 边 (f, i) 表示从 f 到 i].

(a) 在这个网络中寻找最大 $L-W$ 流和最小 $L-W$ 切割.

(b) 通过观察, 寻找另一个最大 $L-W$ 流.

(c) 做一个流入途径错误的流, 它产生一个饱和 $L-W$ 切割, 其容量大于这个流 (类似于图 4.7e 的情况). 现在, 运用本节所给的增广流算法.

(d) 另外, 设每条边上的第二数值是这条边上的流的下界. 尝试着寻找同时满足两个限制的 $L-W$ 流.

5. 删除 4.2 节图 4.2 中连接 y 的各顶点 p, 把剩余边上的数值看成容量 (边仍然是无向的). 使用例 5 中提到的流算法的无向版本.

(a) 寻找从 b 到 j 的最大流和最小 $b-j$ 切割.

(b) 构建使边 (k, l)(在任意方向上) 饱和的 $b-j$ 流, 然后运用该算法.

(c) 把边上的数值看成下界, 修改算法来寻找最大 $b-j$ 流, 使用整个网络 (从 a 到 y 的整个网络). 记住, 没有进入 b 的流, 也没有从 j 出来的流.

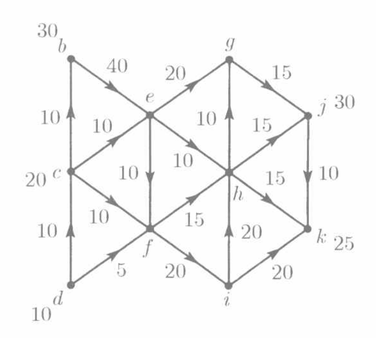
6. 在图 4.4 中存在满足需求的流吗?

7. 假设图 4.4 中的顶点 b, c, d 有无限制供应. 可以有多少流被送入集合 $\{h, i, j\}$? 解释你的模型.

8. 在右图的网络中, 顶点 b, c, d 的供应量分别为 30, 20, 10, 顶点 j 和 k 的需求量分别是 30 和 25.

(a) 如果可能, 寻找满足需求的流.

(b) 反转边 (h, g) 的方向, 重复部分 (a).



9. 解决例 6 中的信使问题.

10. 假设例 6 中的每条边最多可供 3 名信使使用. 此时, 可以运送多少名信使? 在这种修改形式下的答案总是原始问题答案的 3 倍吗?

11. **(a)** 不计 4.1 节练习 3 中的边数, 从 L 到 W 的边不相交路径的最大集合的大小是多少?
(b) 在每条边最多可被 5 条路径使用的条件下, 从 L 到 W 的路径的最大集合的大小是多少?

12. 对于图 4.8a, 假设中间顶点 b, c, d, e, f, g 中的每一个最多可经过 5 个单位的流. 现在, 寻找这个修改后的网络中的最大流和相关的最小 $a-z$ 切割.

13. 对于图 4.4, 假设中间顶点 b, c, d, e, f, g 中的每一个最多可经过 20 个单位的流. 现在, 寻找这个修改后的网络中的最大流.

14. 对于图 4.10, 从 a 到 z 的顶点不相交路径的最大集合的大小是多少?

15. 求解例 7 中的动态流问题.

16. 对于例 7, 假设火车不是每天运行. 设火车在星期一、星期三和星期五从 a 和 c 出发, 而在星期二、星期四和星期六从 b 和 d 出发. 在一周内, 从星期一到星期天, 在那个网络中从 a 到 z 可以运送多少辆汽车?

17. 对于图 4.5, 假设穿过每一条边需要一天时间. 在这个网络中, 5 天内从 a 到 z 可以移动多少个单位?

18. 在定理 3 的证明中, 证明 $f^0 = mf_K$ 满足条件 (b) 的第二个部分.

19. **(a)** 证明: 如果对于某个 x, y, 有向网络包含边 $(x^{\rightharpoonup}, y)$ 和 $(y^{\rightharpoonup}, x)$, 那么增广流算法从不会做使得在这两个边中同时有流通过的赋值.
(b) 如果增广流算法的步骤 (2) 在检查进入边之前检查出去边, 那么边 $(x^{\rightharpoonup}, y)$ 和 $(y^{\rightharpoonup}, x)$ 能得到流吗?

20. **(a)** 重述增广流算法, 使得从 z 开始赋标签, 而且 "反向工作" 到 a.
(b) 重述无向网络的增广流算法 (参见例 5). 概述这一算法的证明.

21. 证明: 在增广流算法的步骤 2a 和 2b 中, 用于给顶点赋标签的边的集合形成以 a 为根的树.

22. **(a)** 给出 $a-z$ 流定义的条件 (b) 的一个更弱的替代条件. 这个新条件仍就保证新流从 a 到 z.
(b) 假设条件 (b) 被删除. 存在违反条件 (b) 的最大流吗? 证明或者给出反例.

23. 构建图 4.8a 的网络中的流, 使得它有下述性质:
(a) 它的值 $|f|$ 等于 0, 但并不是所有边都有 0 流;
(b) 它的值等于 2, 但是它不可能被分解成 $a-z$ 流路径的和.

24. **(a)** 证明: 对于网络中的流 f(有向网络或无向网络), 如果 f 不含回路 (circuit-free), 即, 有流的边集合不形成 (有向) 回路, 那么 f 可以分解成为 $a-z$ 流路径之和. (提示: 对 f 的值用归纳法证明.)
(b) 使用这一证明得到分解任意这样的 f 的算法.
(c) 论证任意流 f 可以分解成为 $|f|$ 个 $a-z$ 流路径及一个回路集合.

25. **(a)** 证明: 从一个不含回路的流开始 (参见练习 24), 也许是一个零流, 由增广流算法产生的最大流是一个不含回路的流, 因此它可以被分解成为 $a-z$ 流路径之和.
(b) 使用问题 (a) 和练习 24(b), 寻找例 5 的最大电话呼叫集合的路线.

26. 为了执行增广流算法 (产生增大的流或最小切割) 的一次完整迭代, 在步骤 (2) 中一条边要被检查多少次?

27. 证明: 如果流被分解成为单位流路径, 且如果每一个单位路径穿越给定的饱和切割一次, 那么这个流是最大的.

28. 无向图 G 的一个**切割集**是这样的边集合 S: 这些边的删除使 G 不再连通, 但是 S 的任何真子集都不能使 G 不连通. 证明: 在无向流网络中, 每一个分离 a 和 z 的切割集是一个 $a-z$ 切割, 且每一个最小 $a-z$ 切割是一个切割集.

29. 设 G 是连通无向图, 且 a, b 是 G 的任意两个顶点.

(a) 证明: a 和 b 之间存在 k 条没有共同边的路径, 当且仅当每一个 $a-b$ 切割至少有 k 条边;

(b) 证明: a 和 b 之间存在 k 条顶点不相交路径, 当且仅当使 a 和 b 不连通的每一个顶点集合至少有 k 个顶点.

30. 正如在例 6 中所提到的那样, 我们可以用另一个多重流网络 N' 来对流网络 N 建模, 在 N' 中每条边都有单位容量; N' 的项点数与 N 相同, 而且对于 N 的每条边 e, 在 N' 中都存在 $k(e)$ 条平行于 e 的边. 因为 N' 中边上的流取值为 0 或 1, 所以可以取消 N' 的容量, 得到一个多重图 G'. N' 中的一个流正好是 G' 的一个边 (有流的边) 子集.

(a) 刻画对应于 N' 中一个 $a-z$ 流的 G' 的边子集.

(b) 使用 G' 重述增广流算法.

(c) 使用多重图模型, 证明: N' 的任意 $a-z$ 流包含 $|f|$ 条 $a-z$ 流路径.

(d) 使用这个多重图模型并假设有练习 29(a) 的结果, 证明推论 3a(最大流–最小切割定理).

31. 假设有向网络的边上的数值表示流的下界. 描述一个减少流算法. 概述这一算法的证明并推导出相应的推论 3a.

32. **(a)** 解释如何把练习 31 中的算法运用于二部图, 以寻找与这个图的每一个顶点都相连的最小边集合.

(b) 寻找 4.4 节图 4.12 中的二部图的一个这样的边集合.

33. 假设我们已知有向网络每条边 e 上的流的上界 $k_1(e)$ 和下界 $k_2(e)$, 且给定一个 (满足这些限制) 的适当流.

(a) 修改增广流算法, 使得可以使用这个算法从给定的可行流出发构建一个最大流.

(b) 证明: 在所有的 $a-z$ 切割 $(P, \bar{P})$ 中, 上面的最大流的值等于 $k_1(P, \bar{P})-k_2(\bar{P}, P)$ 的最小值, 其中 $k_1(S, \bar{S})$ 和 $k_2(S, \bar{S})$ 分别是切割 $(S, \bar{S})$ 中所有边的上界和下界之和.

34. 考虑如例 1 所给出的有供应和需求的有向网络. 设 $z(P)$ 是集合 P 中顶点的总需求, $a(P)$ 是 P 中顶点的总供应.

(a) 证明: 需求能够得到满足, 当且仅当对于所有集合 P, $z(P)-a(P) \leqslant k(\bar{P}, P)$. (提示: 扩展例 1 中的推理.)

(b) 证明: 供应可以全部用尽, 当且仅当对于所有 P, $a(P)-z(P) \leqslant k(P, \bar{P})$.

35. 假设例 7 的图 4.11a 中的边是无向的. 如何构建一个静态流模型来模拟这个动态流问题, 使得这个新网络的最大静态流将对应于最大动态流? 有什么困难吗?

36. 假设一个无向流网络是一个可平面图, 且 a(在左侧) 和 z(在右侧) 都在包围这个网络的无

界区域内. 画出从 a 无限伸长到左侧的边, 以及从 z 无限伸长到右侧的边, 给这些边无限的容量. 这样就把一个无界区域分成两个无界区域, 即上无界区域和下无界区域. 现在, 构建这个可平面网络的对偶网络 (参见 1.4 节), 并给每一条对偶边分配它所穿越的原始边的容量.

(a) 证明: 在对偶网络中, 从上无界区域的顶点到下无界区域的顶点的路径对应于原来网络的一个 $a-z$ 切割. 因此, 对偶网络中这样的最短路径是原始网络中的最小 $a-z$ 切割.

(b) 画出图 4.9 的网络的对偶网络, 并寻找对应原始网络中最小 $a-z$ 切割的最短路径 (使用 4.1 节的算法).

37. 假设无向网络 N 是一个可平面图, 且 a(在左侧) 和 z(在右侧) 都在包围这个网络的无界区域上. 考虑下面的流路径构建的启发式方法. 从 a 开始构建一个 $a-z$ 流路径: 在每一个顶点 x 处, 从用于进入 x 的边开始, 以顺时针方向选择第一条未饱和边.

(a) 把这一启发式方法运用于图 4.9 的网络来寻找最大流.

(b) 证明: 反复使用这一启发式法可以产生 N 的一个最大流.

38. 对于带无理数容量的网络, 证明: 如果该网络的顶点是有序的 (标有下标), 而且增广流算法在步骤 (2) 要扫描的下一个顶点是带最小下标的已赋标签顶点, 那么对这个网络有限次反复使用增广流算法可以产生一个最大流.

39. 编写寻找有向网络中最大流的程序 (网络是输入数据).

40. 编写寻找无向图中两个给定顶点间使下面描述为真的路径的最大数量的程序:

(a) 这些路径没有公共边; **(b)** 这些路径没有公共顶点.

41. 编写这样的程序: 给定值为 k 的 $a-z$ 流的网络, 这个程序从这个流中提取 k 条从 a 到 z 的单位流路径.

42. 编写这样的程序: 给定连通图中的一对顶点 a 和 b, 这个程序寻找其删除使 a 和 b 不再连通的顶点的最小集合 (参见练习 29).

4.4 算法上的匹配

本节把网络流运用于匹配理论. 回想一下, 二部图 $G=(X,Y,E)$ 是这样的无向图: 有两个特殊的顶点集合 X, Y, 且所有边都形如 (x, y), 其中 $x \in X$, $y \in Y$. 参见图 4.12. 二部图是匹配问题的自然模型. 我们设 X 和 Y 是要匹配的两个集合, 且边 (x, y) 代表可以匹配到一起 (配对) 的元素对.

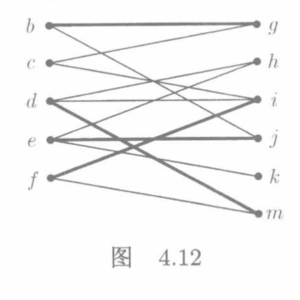


图 4.12

二部图的一个**匹配**(matching) 是**独立边**(没有共同端顶点) 的集合. 图 4.12 中的粗边组成一个匹配. X 匹配是包含 X 中所有顶点的匹配. **最大匹配**(maximum matching) 是有最大可能大小的匹配. 正如对网络流那样, 我们并不总能通过简单地给非最大匹配加入更多的边来得到二部图的最大匹配. 图 4.12 所示的匹配不能再增加, 即使它不是最大的.

一个典型的匹配问题是舞会上适当的男孩与女孩的配对问题, 或者一对一地把某项工作分配给接受过这项工作训练的工人. 与此密切相关的问题是在一组子集中寻找**不同代表的集合**(set of distinct representative). 我们需要从每个子集中挑选出一个元素, 且不使用任意元素两次. 在二部图模型中, 我们为每个子集做一个 X 顶点, 为每个元素做一个 Y 顶点, 而且当元素 y 在子集 x 中时有边 (x, y). 反过来, 任意匹配问题都可以模型化为不同代表集合问题.

我们采用 4.3 节例 1 中的巧妙修改来把匹配问题转化成为网络流问题. 在每一个 X 顶点处指定 1 的供应, 在每一个 Y 顶点处指定 1 的需求. 从 X 到 Y 的各边的容量可以是任意大的正整数, 但是为了方便起见, 假设这些容量是 ∞. 我们还假设边从 X 到 Y 是有向的. 现在, 运用 4.3 节例 1 中的技术, 其中源点 a 通过一条单位容量边与每一个 X 顶点连通, 汇点 z 通过来自每一个 Y 顶点的单位边与它们连通.

我们称这样的网络为**匹配网络**(matching network). 参见图 4.13. 用于 $a-z$ 流的 $X-Y$ 边组成一个匹配. 最大流是最大匹配. 来自源点 a 且使所有边饱和的流对应于一个 X 匹配.

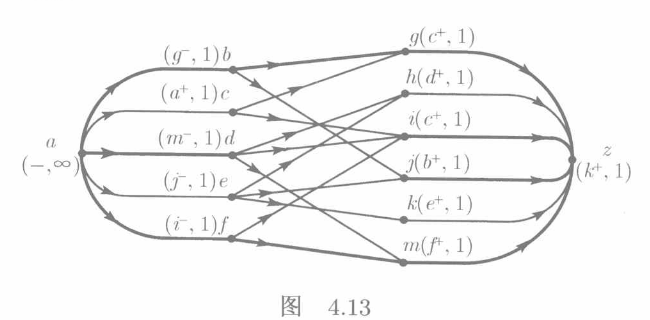


图 4.13

例 1 最大匹配

假设图 4.12 的二部图代表男孩 b, c, d, e, f 与女孩 g, h, i, j, k, m 的可能配对. 在图 4.12 中用粗边表示临时匹配. 尽管这一匹配不能扩展, 但是我们仍想知道完全的 X 匹配是否可能.

正如在之前的流问题中所出现的情况那样, 在这一匹配中我们犯了错误, 因此现在需要重新做某些分配. 把这一匹配转化成在相应的匹配网络中的对应流: 图 4.13 的黑边有 1 流, 其他边有 0 流. 现在, 运用增广流算法. 参见图 4.13.

从 a 开始, 唯一非饱和的出去边是 $(a^{\rightarrow}, c)$, 因为 c 是唯一不包含在初始匹配中的 X 顶点. 我们给 c 赋标签 $(a^+, 1)$. 从 c 开始, 可以给两个与 c 相邻的 Y 顶点 g 和 i 赋标签 $(c^+, 1)$. 在 g 处, 一条到 z 的出去边是饱和的. 于是, 一定存在一个来自未赋标签顶点 b 的到 g 有正流的进入边, 比如说是 $(b^{\rightarrow}, g)$. 给 b 赋标签 $(g^-, 1)$. 通过类似的推理, 从 i 出发, 我们给 f 赋标签 $(i^-, 1)$. 在 b 和 f, 可以给不与 b

和 f 匹配的 Y 顶点赋标签. 给 j 赋标签 $(b^+, 1)$, 给 m 赋标签 $(f^+, 1)$.

概括至此所赋的标签, 我们正在尝试寻找 c(当前唯一未匹配的 X 顶点) 的匹配. c 的可能匹配是 g 和 i, 但是, 做任意的匹配, 我们必须切断当前 g 与 b 或者 i 与 f 的匹配, 在这个网络中, 重新把 b 或者 f 与其他 Y 顶点匹配, b 可以重新与 j 匹配, 或者 f 重新与 m 匹配. 继续这一推理, 为了重新把 b 与 j 匹配要求我们删除当前的 j 与 e 的匹配, 并寻找 e 的新匹配. 或者, f 与 m 的匹配要求我们必须删除 m 当前与 d 的匹配, 并为 d 寻找新的匹配. 现在, 或者 d 与 (当前没有匹配的)h 匹配, 或者 e 与 (当前也没有匹配的)k 匹配. h 和 k 没有匹配的事实表明, 从 b 和 k 中的一个出发我们可以给 z 赋标签. 参见图 4.13 中的标签.

如果给 z 赋标签 $(k^+, 1)$, 由增广流算法所描述的增广流链是 $a-c-g-b-j-e-k-z$. 这一增广流链表明我们把边 $(c^{\rightarrow}, g)$ 加入到匹配中, 删除 $(b^{\rightarrow}, g)$, 把边 $(b^{\rightarrow}, j)$ 加入, 并删除边 $(e^{\rightarrow}, j)$, 把边 $(e^{\rightarrow}, k)$ 加入. 新流对应于匹配 $b-j$, $c-g$, $d-m$, $e-k$, $f-i$. 再运用这一算法将只给 a 赋标签. 离开 a 的边集合形成一个(饱和的) 最小 $a-z$ 切割, 这表明我们有一个 X 匹配. □

观察发现, 在匹配问题中增广流算法的动作可以如下描述: 从 X 中一个未匹配顶点 x_1 开始, 进入一个未匹配边 (当前在匹配中没有使用的边) 到达一个已匹配的 Y 顶点 y_1, 然后从 y_1 开始沿着一个匹配边返回到一个已匹配的 X 顶点 x_2, 然后从 x_2 开始进入一条未匹配边来到一个已匹配顶点 y_2, 一直这样下去, 直到未匹配的边 (x_k, y_k) 从一个已匹配的顶点 x_k 出发到达一个未匹配的 Y 顶点 y_k. 总之, 从一个未匹配的 X 顶点开始, 在寻找一个没有匹配的 Y 顶点的过程中, 我们创建未匹配和已匹配边交替的奇长度路径 L. 给定一个这样的路径, 通过交换 L 上的匹配边与未匹配边的角色, 我们得到一个新的更大的匹配边集合.

我们已经看到, 在二部图中, 匹配类似于网络中的流. 那么二部图中的什么对应于 $a-z$ 流呢? 事实上, 可以把注意力局限于有限容量的 $a-z$ 切割. 对应的概念是**边覆盖**(edge cover), 即, 一个顶点集合 S, 使得每一条边都有一个在 S 中的端顶点. 回想一下, 我们已经在 1.1 节例 4 的街道监控问题中遇到过边覆盖了.

引理 设 $G=(X,Y,E)$ 是二部图, 且设 N 是与 G 相关的匹配网络. 对于任意 (可能是空的) 子集 $A\subseteq X$ 和 $B\subseteq Y$, $S=A\cup B$ 是边覆盖当且仅当 $(P, \bar{P})$ 是 N 中的有限容量 $a-z$ 切割, 其中 $P=a\cup(X-A)\cup B$. 用 P 表示, $S=(\bar{P}\cap X)\cup(P\cap Y)$. 因此, $|S|=k(P,\bar{P})$.

证明 有限容量 $a-z$ 切割不能包含 X 和 Y 之间的边 (其容量是 ∞). 于是, 一定存在集合 $A\subseteq X$ 和 $B\subseteq Y$, 使得有限容量切割 $(P, \bar{P})$ 由形如 $(a^{\rightarrow}, x)$ 和形如 $(y^{\rightarrow}, z)$ 的边组成, 其中 $x\in A$, $y\in B$. 这些边阻塞所有流 (因此, 是 $a-z$ 切割), 当且仅当任意的 $X-Y$ 边从某个 $x\in A$ 开始, 或结束于某个 $y\in B$, 即当且仅当 $S=A\cup B$ 是边覆盖. 用 A 和 B 表示, $P=a\cup(X-A)\cup B$; 用 P 表示, $A=\bar{P}\cap X$, $B=P\cap Y$. 参见图 4.14. 而且, $|S|=k(P,\bar{P})$, 因为 $(P, \bar{P})$ 中的边都有单位容量. ◇

现在, 我们证明二部图匹配中两个著名的定理.

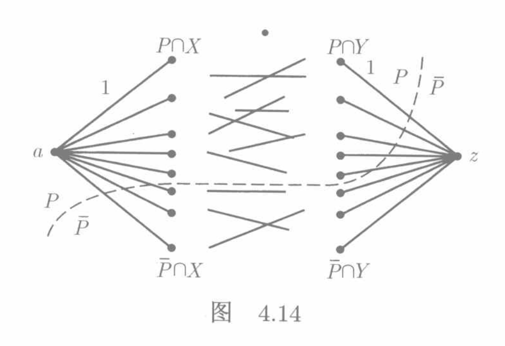


图 4.14

定理 1(König-Egevary, 1931 年) 在二部图 $G=(X,Y,E)$ 中, 最大匹配的大小等于最小边覆盖的大小.

匹配对应于相关匹配网络中的流, 根据引理, 边覆盖对应于有限容量 $a-z$ 切割. 定理 1 就是最大流–最小切割定理 (4.2 节推论 3a) 的二部图重述.

在下面的定理中, A 的**延伸**(range)$R(A)$ 表示与 A 中某个顶点相邻的顶点的集合.

定理 2(Hall 婚配定理, 1935 年) 二部图 $G=(X,Y,E)$ 有 X 匹配, 当且仅当对于每一个子集 $A\subseteq X$, 有 $|R(A)|\geqslant|A|$.

证明 上面的条件是必要的, 因为如果 $|R(A)|<|A|$, 那么 A 中顶点的匹配显然是不可能的, 所以 X 匹配是不可能的.

接下来证明: 如果对于所有的 $A\subseteq X$, 有 $|R(A)|\geqslant|A|$, 那么 G 有 X 匹配. 根据定理 1, X 匹配存在当且仅当对于每一个边覆盖 S, S 的大小 $|S|\geqslant|X|$. 如果 S 只包含 X 顶点, 即 $S=X$, 那么这一结果是显然的. 必须证明无法通过删除某些 X 顶点, 并在它们的位置上使用更少的 Y 顶点来使 S 更小. 如果 A 是不在 S 中的 X 顶点的集合, 那么为了覆盖 A 与 $R(A)$ 之间的边, $S\cap Y$ 一定包含 $R(A)$ 中的顶点. 于是, 我们有:

$$|S|=|S\cap X|+|S\cap Y|=|X|-|A|+|S\cap Y|\geqslant|X|-|A|+|R(A)|$$

因为 $|R(A)|\geqslant|A|$, 所以 $|X|-|A|+|R(A)|\geqslant|X|$, 所以有 $|S|\geqslant|X|$, 这就证明了存在大小为 X 的匹配. ◇

定理 1 和定理 2 是丰富的匹配理论的出发点 (例如, 参见练习 18 和练习 23).

例 2 A 国想要监视来自 X 国和 Y 国的外交官在其领土内所举行的所有会面. 我们知道哪些外交官彼此有可能会面. 可以使用二部图来表示这种可能会面关系. A 国不可能给 X 国和 Y 国的每名外交官都安排间谍. 它想要寻找最小的外交官集合 S, 使得每一次可能的会面都包含 S 中的一名外交官.

A 国雇用这门课程的一名研究生, 他立即看出这一最小监视问题实际上是最小边覆盖问题. 因此, 可以通过使用增广流算法寻找相关匹配网络中的最小 $a-z$ 切割, 并从这个 $a-z$ 切割出发, 使用引理来得到一个最小边覆盖, 从而解决这一问题. □

例 3 假设有 n 名员工和 n 项工作, 每名员工能胜任 k 项工作. 每项工作有 k 名胜任的员工. 有可能给每名员工分配他能做的 (不同) 工作吗?

我们使用二部图 $G=(X,Y,E)$ 对这一问题建模, 其中 X 顶点代表员工, Y 顶点代表工作. 注意, 每个顶点有度 k. 这样一来, 这一问题变成了, 存在 X 匹配吗? 根据定理 2, 可以如下推导出肯定的答案: 设 A 是 X 的任意子集. 因为每个顶点的度为 k, 所以有 $k|A|$ 条边离开 A. 因为对于 Y 中任意顶点, 至多有 k 条边可以到达这一顶点, 因此 $R(A)$ 至少有 $|A|$ 个顶点. 于是根据定理 2, 存在 X 匹配. □

在本节的最后, 我们看一下匹配网络在体育问题中的应用, 这一问题是确定赛季末某个队是否还有成为赛季冠军的 (数学意义上的) 机会. 这一网络模型和相关的分析是由 Schwartz 提出的, 见文献 [4].

例 4 竞争淘汰

假设我们知道在一次体育竞赛中到目前为止每个队已经胜了多少场比赛, 以及每一对参赛队间还要进行多少场比赛. 我们想要知道是否存在某种方案, 在这一方案下, 某个队能够以最多胜利场次完成这一赛季的比赛.

考虑有 4 个队的具体例子, 它们是 Bears、Lions、Tigers 和 Vampires, 并分别记有下面的胜利场数及剩余比赛场数. 参见图 4.15. 假设我们想要知道, (目前有最少胜利场次的)Bears 是否有可能以最多的胜利场次结束本赛季. 现在问题是, 即便假设 Bears 能够赢得所有剩余比赛, 最终以赢 23 场的成绩结束赛季, 其他各队间的比赛结果总有可能使其中的某队比 Bears 赢的场次多. 因此, 问题的关键就是寻找下面这一问题的答案:

是否存在指定其他各队间比赛的胜负, 使得每个队伍在比赛结束时最多胜 23 场?

队伍	目前赢得的场次	剩余的比赛场次	与Bears比赛	与Lions比赛	与Tigers比赛	与Vampires比赛
Bears(t_1)	16	7	—	2	2	3
Lions(t_2)	22	7	2	—	3	2
Tigers(t_3)	20	8	2	3	—	3
Vampires(t_4)	19	8	3	2	3	—

图 4.15

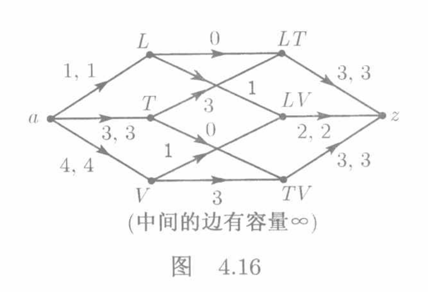


图 4.16

我们通过讨论如何构建一个匹配网络模型来回答上面这个问题. 图 4.16 给出图 4.15 中的数据所组成的网络. X 顶点代表除 Bears 以外的其他队伍. 在图 4.16 中, X 顶点是 L(代表 Lions), T(代表 Tigers), V(代表 Vampires). Y 顶点代表所有不同的队伍配对. 这里, 配对为 LT, LV, TV.

从源点 a 到 X 顶点 t_i 的边上的流的大小, 代表在这个赛季的其余比赛中第 i 个队伍 t_i 所赢的场次数. 边 $(a^{\rightarrow}, i)$ 的容量是 $23-w_i$, 其中 w_i 表示到目前为止 t_i

所赢的场次数. 观察发现, 如果队伍 t_i 到目前为止已经赢了 w_i 场, 而且在这个赛季的剩余比赛中它至多赢 $23-w_i$ 场, 那么 t_i 赢的场次数不多于 23, 使得 Bears 成为冠军或并列冠军 (假设 Bears 赢得所有剩余比赛).

$X-Y$ 边 $(t^{\rightarrow}, (t_i, t_j))$ 中的流表示在 t_i 和 t_j 的剩余比赛中, t_i 所赢得的场次数. 可以设 $X-Y$ 的所有边的容量是 ∞. 从 Y 顶点 (t_i, t_j) 到汇点 z 的边的流代表队 t_i 和 t_j 之间将要举行的比赛场次数. $((t_i, t_j)^{\rightarrow}, z)$ 的容量是 t_i 和 t_j 之间安排的剩余比赛的场次数. 顶点 t_i 处的流的守恒性要求进入流 (在剩余赛季中赢的场次数) 等于出去流 (对其余各队伍的将要赢的场次数). 顶点 (t_i, t_j) 处的流的守恒性要求进入流 (t_i 赢 t_j 的场次数及 t_j 赢 t_i 的场次数) 等于出去流 (两个队伍之间要举行的比赛的场次数).

为了得到代表真实的可能方案的流, 我们要求从 Y 到 z 的边都是饱和的, 即 t_i 和 t_j 之间已经进行的比赛场次数等于它们之间事先安排好的比赛场次数. 如果在这样的网络中存在使 $Y-z$ 边饱和的流, 则会有这样一个方案, 通过它, Bears 能够在赛季结束时胜利的场次数最多. 图 4.16 给出饱和 $Y-z$ 边的流, 说明了 Bears 是如何以最多的胜利场次数完成比赛的 (实际上 4 个队都是并列第一). □

选读 现在, 我们扩展前面的体育网络模型, 给出某个队伍 t^* 成为赛季冠军的必要条件和充分条件. 对于队伍的子集 S, 设 $w(A)$ 表示 S 中所有队伍目前赢得的总场数, 并设 $r(S)$ 是 S 中所有队伍对之间要举行的剩余比赛的总场数. 观察发现, 对于队伍的任意子集 S, 在赛季结束时, S 中的各队的胜利场次总数至少是 $w(S)+r(S)$(因为当 S 中两个队伍之间相互比赛时, 其中之一一定赢). 如果 t^* 赢得所有其余比赛, 设 W^* 是 t^* 最终赢得的场次数. 能够保证 t^* 胜利的场次数不是最多的一个限制是: 如果不包含 t^* 的任意子集 S 所赢得的场次数的平均数超过 W^*, 即

$$\text{对于不包含}t^*\text{的任意子集}S\text{有}:\{w(S)+r(S)\}/|S|>W^* \qquad (*)$$

我们将证明限制 $(*)$ 是刻画 t^* 被淘汰的充分条件和必要条件. 如果最小切割是 $Y-z$ 边的集合, 那么存在一个饱和所有 $Y-z$ 边且决定下面方案的流: 在这个方案下, t^* 赢得的场次数最多 (正如在上面例子中所描述的那样). 假设存在一个有最小容量的 $a-z$ 切割 $(P, \bar{P})$. 现在, 我们证明某个子集 S 满足 $(*)$.

$(P, \bar{P})$ 不能包含任何从 X 到 Y 的无限容量边, 所以它只能是由 $a-X$ 边和 $Y-z$ 边组成的. 回想一下, 边 $(a^{\rightarrow}, t_i)$ 的容量是 W^*-w_i. 于是, 如果 $A=\bar{P}-\cap X$, 那么 $(P, \bar{P})$ 中的 $a-X$ 边的容量是 $|A|W^*-w(A)$. 可以证明: 如果 t_i 和 t_j 都在 $\bar{P}$ 中, 那么 (t_i, t_j) 必定在 $\bar{P}$ 中 (详细证明留作练习). 于是, $(P, \bar{P})$ 中的 $Y-z$ 边的容量是 $\sum\limits_{(t_i,t_j)\not\subset A} r_{ij}$, 其中 r_{ij} 表示边 $((t_i, t_j)^{\rightarrow}, z)$ 的容量 (它等于 t_i 和 t_j 之间要举行的剩余比赛场次数), 我们对所有这样的 (t_i, t_j) 对求和, 其中 t_i 或者 t_j 至少有一个不在 A 中. 因此 $k(P,\bar{P})=[|A|W^*-w(A)]+\sum\limits_{(t_i,t_j)\not\subset A} r_{ij}$.

假设 $k(P,\bar{P})$ 小于所有从 Y 到 z 的边的总容量 $\sum r_{ij}$, 那么有:

$$
\begin{aligned}
&|A|W^* - w(A) + \sum_{(t_i,t_j)\not\subset A} r_{ij} < \sum r_{ij} \\
&\qquad \Rightarrow |A|W^* - w(A) < \sum r_{ij} - \sum_{(t_i,t_j)\not\subset A} r_{ij} = \sum_{(t_i,t_j)\subset A} r_{ij} = r(A) \\
&\qquad \Rightarrow |A|W^* < w(A) + r(A) \Rightarrow W^* < \{w(A) + r(A)\}/|A|
\end{aligned}
$$

最后的不等式就是我们要证的 $(*)$, 其中子集是 A. A 中队伍平均赢得的场次数超过 t^*, 所以 t^* 不可能是赛季冠军.

练　习　4.4

练习概况　练习 1~14 是有关匹配网络的. 练习 15~24 拓展理论.

1. Ann、Diana 和 Lolita 喜欢 Bill, Bobbie、Carol 和 Lolita 喜欢 Fred, Ann、Bobbie 和 Lolita 喜欢 George, Carol 和 Lolita 喜欢 John, Diana 和 Lolita 喜欢 Larry. 我们想要把每个女孩与她喜欢的男孩组对. (以女孩作为左边的顶点.)

(a) 建立相关的匹配网络, 并最大化它的流来解决这一问题.

(b) 做一个对应于部分配对的流, 这个部分配对包含 Bill 与 Diana、Fred 与 Carol 及另两个 (由读者自己选定的) 配对. 现在, 运用流增广算法增加这个匹配, 使其成为一个完全匹配.

2. 假设有 5 个委员会: 委员会 A 的成员是 a, c, e; 委员会 B 的成员是 b, c; 委员会 C 的成员是 a, b, d; 委员会 D 的成员是 d, e, f; 委员会 E 的成员是 e, f. 我们希望从每一个委员会派出一名不同的代表出席一次会议.

(a) 建立相关的匹配网络, 并最大化它的流来解决这一问题.

(b) 做下面的对应于部分委派的流: A 派出 e, B 派出 b, C 派出 a, D 派出 f. 现在运用流增广算法增加这个匹配.

3. 重复练习 1, 但这次 Bill 有 5 次约会, Fred 有 4 次约会, George 有 3 次, John 有 5 次, Larry 有 3 次, 而同时, Ann 有 4 次, Bobbie 有 3 次, Diana 有 5 次, Carol 有 4 次, Lolita 有 4 次. 合适的配对可以在一起有任意次约会. 使用网络建模来寻找可能的配对集合.

4. 重复练习 1, 但这次我们想要为每一个男孩 (与两个不同的女孩) 配对两次, 并为每一个女孩配对两次. 使用一个适当的网络流建模来寻找这样的配对.

5. 假设有 6 所大学, 这一年每所大学将产生 5 位数学博士, 有 5 所学校分别雇用 7, 7, 6, 6, 5 名数学博士. 任何学校都不会从任意给定的大学雇用 1 名以上博士. 所有博士都能得到一份工作吗? 请解释.

6. 在例 4 中, 如果 Vampires 能够赢得所有剩余比赛, 那么它有可能成为冠军 (或并列冠军) 吗? 构建适当的网络模型.

7. **(a)** 对于下面的剩余比赛表, 如果 Bears 赢得所有剩余的比赛, 它有可能成为并列冠军或唯一的冠军吗? 构建适当网络模型, 并在可能时给出所需的流. 在你所创建的网络里用一条可行的流回答这一问题, 或者解释为什么这是不可能的. 如果有与 Bears 一起获得并列冠军的队伍, 他们可能是谁?

(b) Bears 有可能成为唯一的冠军吗?

队伍	目前赢得的场次	剩余的比赛场次	与Bears比赛	与Lions比赛	与Tigers比赛	与Vampires比赛
Bears	21	7	—	1	2	4
Lions	26	6	1	—	3	2
Tigers	26	7	2	3	—	2
Vampires	22	8	4	2	2	—

8. 对于下面的剩余比赛表, 如果 Vikings 赢得所有剩余比赛, 那么它有可能成为并列冠军或唯一的冠军吗? 构建适当的网络模型和流. 在你所创建的网络中用一条可行流回答这一问题. 或者解释它为什么不可能. 如果有与 Vikings 一起得到并列冠军的队伍, 那么它们可能是谁?

队伍	目前赢得的场次	剩余的比赛场次	与Vikings比赛	与Huns比赛	与Romans比赛	与Mongols比赛
Vikings	22	6	—	2	2	2
Huns	27	6	2	—	2	2
Romans	26	6	2	2	—	2
Mongols	25	6	2	2	2	—

9. 对于下面的剩余比赛表, 如果 Bears 赢得所有剩余比赛, 那么它有可能成为并列冠军或唯一的冠军吗? 构建适当的网络模型和流. 在你所创建的网络中用一条可行流回答这一问题. 或者解释它为什么不可能. 如果有与 Bears 一起得到并列冠军的队伍, 那么它们可能是谁?

队伍	目前赢得的场次	剩余的比赛场次	与Bears比赛	与Lions比赛	与Tigers比赛	与Vampires比赛
Bears	18	8	—	1	3	4
Lions	25	6	1	—	2	3
Tigers	24	7	3	2	—	2
Vampires	22	9	4	3	2	—

10. 对于下面的剩余比赛表, 如果 Bears 赢得所有剩余比赛, 那么它有可能成为并列冠军或唯一的冠军吗? 构建适当的网络模型和流. 在你所创建的网络中用一条可行流回答这一问题. 或者解释它为什么不可能. 如果有与 Bears 一起得到并列冠军的队伍, 那么它们可能是谁?

队伍	目前赢得的场次	剩余的比赛场次	与Bears比赛	与Lions比赛	与Tigers比赛	与Vampires比赛
Bears	20	6	—	1	2	3
Lions	25	6	1	—	3	2
Tigers	25	6	2	3	—	1
Vampires	22	6	3	2	1	—

11. 计算机约会服务中心有 n 个男孩和 n 个女孩. 计算机做了 nm 个配对, 使得每个男孩与 m 个不同女孩约会, 每个女孩与 m 个不同男孩约会 $(m < n)$.

(a) 证明: 总有可能在 m 个夜晚安排 nm 次约会, 即, 这些配对可以分割成为 m 个完全

配对的集合.

(b) 证明: 在问题 (a) 中, 无论前 k 个完全配对 $(0 < k < m)$ 如何选择, 这一分割总可以成功.

12. 我们想要构建项为非负整数的 $n \times m$ 阶矩阵, 使得第 i 行上的所有项之和等于 r_i, 第 j 列上的所有项之和等于 c_j. 显然 r_i 的和一定等于 c_j 的和.

(a) 对 r_i 和 c_j 还需要施加什么其他限制 (如果有的话), 以便保证这样的矩阵存在?

(b) 构建这样的一个 5×6 阶矩阵, 其中各行的和分别等于 20, 40, 10, 13, 25, 而各列的和都等于 18.

13. 有一组人, 每个人都是委员会的一个子集的成员. 另外, 他们都毕业于 3 所不同大学. 对于这一 "不同代表" 问题的扩展, 为每个委员会找一个代表, 且附加保证来自每一所大学的人数占代表的 $\frac{1}{3}$ 的限制. 描述如何构建一个网络来对这一问题建模. (假设委员会的数量 m 是 3 的倍数.)

14. (得自 Bacharach) 给定一个具有数值项的 $n \times n$ 阶矩阵, 它的所有项都有一位十进制小数 (例如, 13.3). 我们想要以某种方式四舍五入这些项, 使得每一列 (及行) 中四舍五入后的项的和是原来列 (行) 上的项的和的四舍五入值. 例如, 如果第一列有项 2.5, 6.4, 5.7, 它们的和等于 16.6, 而这些项四舍五入后的和等于 16 或者 17.

(a) 描述如何构建这一问题的匹配类型网络流模型. 设 X 顶点代表列, Y 顶点代表行. 注意, 这一模型的每个边都有下界和上界.

(b) 对下面矩阵使用 (a) 中构建的网络模型, 寻找所需的四舍五入:

$$\begin{matrix} 4.5 & 7.5 & 2.5 \\ 6.8 & 4.3 & 5.7 \\ 3.6 & 1.6 & 4.3 \end{matrix}$$

15. 若在有向图 $G = (V, E)$ 中, 存在经过每个顶点一次的回路或者顶点不相交回路, 请给出其必要条件和充分条件. [提示: 做一个二部图 $G' = (X, Y, E)$, 其中 X, Y 是 V 的拷贝, 对于 G 中的每一条边 $(\vec{v_1}, v_2)$, G' 有边 (x_1, y_2), 重述这一问题.]

16. 证明: 二部图 $G = (X, Y, E)$ 有大小为 t 的匹配, 当且仅当对于 X 中的所有 A, $| R(A) | \geqslant | A | + t - | X | = t - | X - A |$. (提示: 把 $|X| - t$ 个新顶点加入到 Y, 并把 Y 的每一个新顶点与每一个 X 顶点相连.)

17. 证明: 每个二部图匹配问题都可以建模成不同代表集合问题.

18. 按照例 4 的分析, 证明: 如果 t_i 和 t_j 在 $\bar{P}$ 中, 那么顶点 (t_i, t_j) 在 $\bar{P}$ 中.

19. 使用 4.3 节的练习 34, 把该练习直接运用于二部图, 而不是增广 $a - z$ 匹配网络, 来证明定理 2.

20. 对完全 Y 匹配, 证明定理 2(不能简单地交换 X 和 Y 的角色). 根据对称性, 我们需要相同的条件, 但是在这一新证明中所选择的集合 A 不同于教材证明中的集合 A.

21. 设 $\delta(G) = \max_{A \subseteq X}(|A| - |R(A)|)$. $\delta(G)$ 称为二部图 $G = (X, Y, E)$ 的亏损(deficiency), 给出不满足定理 2 中条件的情况. 注意, 因为 $A = \phi$ 是 X 的子集, 所以 $\delta(G) \geqslant 0$.

(a) 利用练习 16 证明: G 的最大匹配的大小为 $| X | - \delta(G)$.

(b) 给定大小为 $t = | X | - \delta(G)$ 的最大匹配 [假设 $\delta(G) > 0$], 描述如何使用定理 1 的相关最小边覆盖来寻找使得 $| A | - | R(A) | = \delta(G)$ 的 A.

22. (a) 证明: $G = (X, Y, E)$ 的顶点的最大独立集合 (互不相邻的顶点) 的大小等于 $| Y |$

$+\delta(G)$(参见练习 21). 描述如何寻找这样的独立集合.

(b) 使用问题 (a), 在图 1.3 中寻找这样的独立集合.

23. (得自 J.Hopcroft) 假设对于某个 r, 二部图 G 的每个顶点的度为 2^r. 把 G 的边分割成回路, 并在每条回路中每隔一条边删除一条边. 在新图中重复这一过程 (此时每个顶点的度为 2^{r-1}), 直到得到每个顶点的度为 1 的图为止. 最终图中的边是构成 G 的顶点的一个匹配. 证明: 在每一个后继图中, 这种边到回路的划分都是存在的, 而且可以在与当前图中的边数成正比的大量步数中找到这样的分割.

24. 假设给定任意图的一个部分匹配 (这样的匹配是没有共同端顶点的边的集合).

(a) 证明: 可以沿着一条非匹配与匹配边交替的路径进行交换, 使用这一方法 (按例 1 所述的方法) 的扩展来增加这个部分匹配的大小, 直到它成为一个最大匹配.

(b) 随机挑出图 4.9 的一个部分匹配, 并使用上面的方法得到一个最大匹配.

25. 编写在二部图 (图是输入数据) 中寻找最大匹配和最小边覆盖的计算机程序.

4.5 运 输 问 题

本节运用前几节所探讨的生成树、网络流和匹配的知识来研究最小成本网络流. 为了简化问题, 我们只审视基础图是二部图的匹配网络, 如 4.4 节中的匹配网络. 与匹配问题类似, 这里的边有无限定的容量, 而且顶点有供应和需求, 这里是大于 1 的整数. 称供应顶点为**仓库**(warehouse), 称需求顶点为**商店**(store). 为了进一步简化问题, 假设二部图 $G=(X,Y,E)$ 是完全的, 即每一个仓库 W_i 到每一个商店 S_j 存在一条边 (i, j). 唯一的新地方是边 (i, j) 上存在运输成本 c_{ij}. 我们的目标是寻找把所有物品从仓库运输到商店且使运输成本最小的路径. 参见图 4.17 的样例问题. 这一优化问题称为**运输问题**(transportation problem). 它是运筹学首先研究的优化问题之一.

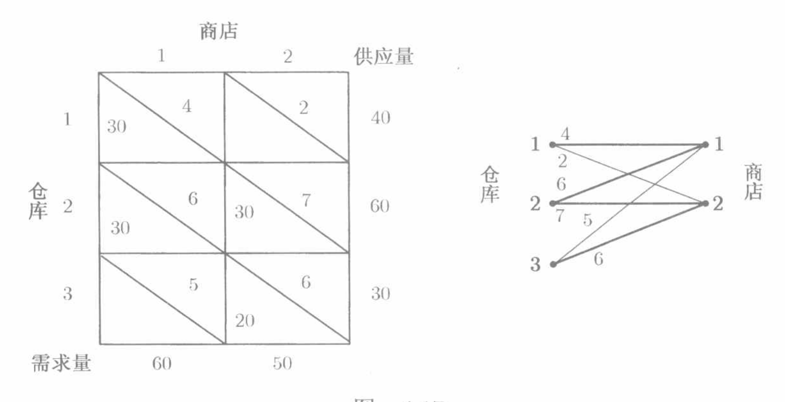


图 4.17

一般地, 仓库 i 的供应量是 s_i, 而商店 j 的需求量是 d_j. 假设所有仓库的总供应量等于所有商店的总需求量. 运输问题的解是指定每条边 (i, j) 上的运输量 x_{ij},

使得每个仓库的总出货不超过这个仓库的供应量, 而且每个商店的总进货等于这家商店的需求量. 目标是在各种解决方案中寻找一个最小成本解. 运输问题的数学描述是:

$$
\begin{aligned}
\min \sum_{i,j} c_{ij}x_{ij} \\
\text{st.} \sum_{j} x_{ij} \leqslant s_i, \forall i \\
\sum_{i} x_{ij} = d_j, \forall j \\
x_{ij} \geqslant 0
\end{aligned}
$$

尽管在这一问题的陈述中并没有出现生成树, 但是事实上生成树是解决这一问题的核心. 最优解中的货运所使用的边的集合形成一棵生成树. 我们寻找最优解的策略是从形成一棵生成树的初始 (非最优) 解出发. 通过反复从当前的生成树中删除一条边, 并用一条 "更好的" 边代替它来寻找更廉价的生成树解, 直到最终得到最优解.

描述运输问题的数据通常用所谓的**运输表**(transportation tableau) 给出. 参见图 4.17 中的样例运输问题的运输表和相关的二部图. (图表示非常杂乱, 所以只适合较小的问题.) 我们将用这个例子展开运输问题求解的分析.

仓库的供应量 s_i 出现在表格的右边, 商店的需求量 d_j 出现在表格的底部. 边 (i, j) 的出货成本 c_{ij} 在表格的项 (i, j) 的右上方. 项 (i, j) 的左下方的数字 x_{ij} 告诉我们从仓库 i 到商店 j 的货运量. [注意, 我们把 (i, j) 等价地称为边或表中的项.] 图 4.17 中的 x_{ij} 是这个运输问题的一个 (非最优) 解. 在图 4.17 中, 我们用粗边表示解所使用的边. 读者可以检查每行 i 中的 x_{ij} 之和是否 $\leqslant s_i$, 以及每列 j 的 x_{ij} 之和是否等于 d_j. 图 4.17 的表格的样例解的成本是 $30\times\$4+30\times\$6+30\times\$7+20\times\$6 = \$630$, 它所用的边是 (1, 1), (2, 1), (2, 2), (3, 2).

为了简化这一问题, 假设供应量之和正好等于需求量之和. 如果供应量超过需求量, 可以创建一个 "虚拟" 商店, 它容纳所有过量的供应. 在图 4.17 中, 总供应量是 130, 总需求量是 110. 在这种情况下, 可以创建一个拥有 $130 - 120 = 20$ 的需求量的虚拟商店, 如图 4.18 中的运输表和图所示的那样. 图 4.17 中的解已经在图 4.18 中得到修改, 使得行 i 的 x_{ij} 之和正好等于 d_i. 沿着任意一条边到这个虚拟商店的单位运输成本等于 0, 因为对于这一技巧, 不涉及平衡供应和需求的实际成本. 因为到虚拟商店的运输成本是 0, 所以从图 4.17 到图 4.18 的转化并没有改变总的运输成本.

现在要证明, 只考虑不包含回路的边集合的解即可. 回想一下, 不包含回路的边集合形成树或一组树.

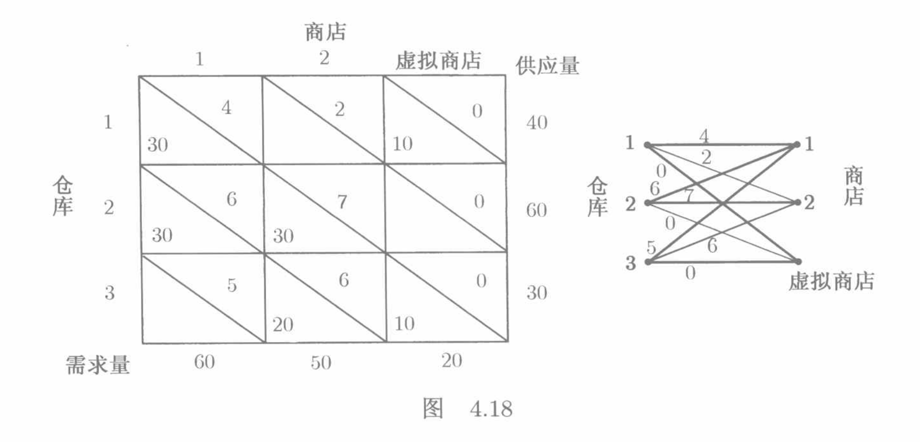


图 4.18

引理 1 *设 S 是一个运输问题的解, 它涉及的边集合 $E(S)$ 包含一条回路. 那么存在另一个成本小于或等于 S 的解 S^*, 其中 $E(S^*)$ 是 $E(S)$ 的不包含回路的子集.*

证明 我们不给出这个引理的形式证明, 而是用一个例子来进行说明, 并解释如何把这个例子扩展到所有运输问题的所有解.

考虑图 4.18 所展示的解 S. S 中的运输所使用的边 (1, 1), (2, 1), (2, 2), (3, 2), (3, 3), (1, 3) 形成一个回路. 可以这样修改 S: 在这条回路的第一条边、第三条边和第五条边 [称为始于 (1, 1) 的这条回路的奇边(odd)] 上把运输量增加 1, 把这条回路的偶边 (其余的边) 上的运输量减少 1. 这些平衡修改保持修改后的 x_{ij} 行和与列和不变, 所以修改后的运输还是一个有效解.

检查一下这个解的原始成本是如何随着 S 的这一修改而发生变化的. 沿着边 (1, 1), (2, 2), (3, 3) 的单位成本增加使成本增加 $4+7+0$(这 3 条边的成本)=11. 沿边 (2, 1), (3, 2) 和 (1, 3) 的单位成本减少使成本减少 $6+6+0=12$. 净改变量是 $11-12=-1$. 因此, 这一修改使解的成本减少 1. 为了通过这种修改得到最大的成本减少, 我们尽可能多地增加这条回路中的奇边上的运输量, 并在这条回路的偶边上减少相应的运输量. 在我们的修改中, 最关键的限制是不能把任意偶边上的运输量减少到 0 以下. 所以, 我们检查 S 中偶边的运输量. 边 (1, 3) 上的数值 10 最小. 所以可以把奇边上的运输量增加 10, 而偶边上的运输量减少 10. 注意, 通过把边 (1, 3) 上的运输量降到 0, 我们把这条边从用于新解的边子集中删除. 因此, 新解不再有回路. 图 4.19 给出这个新的运输表. 读者应该检查图 4.19 的新解的成本是$620, 如期望的那样, 这个成本比图 4.18 的解的成本减少 10.

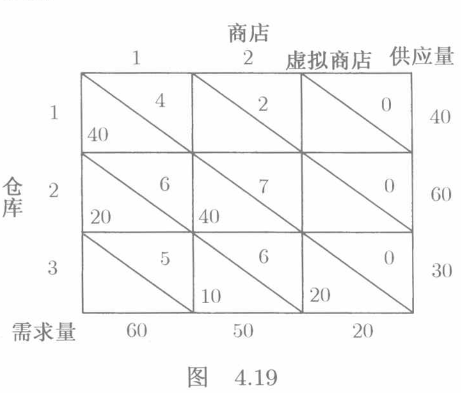


图 4.19

如果我们在奇边把运输量增加 1, 在偶边把运输量减少 1 的修改实际上使成本增加, 那么就要反过来使用这一策略, 尽可能多地减少奇边的运输量, 并相应地增加偶边的运输量, 直至有一条奇边的运输量为 0. 同样, 如果我们的修改没有使成本发生变化, 那么遵循原来增加奇边运输量并尽可能多地减少偶边上出货的策略, 来得到有相同成本的新的无回路解.

因为运输问题的图是二部图, 根据 1.3 节定理可知, 这个图中的每一条回路都有偶长度, 所以, 任意回路的边可以分割成奇边和偶边, 如在前面例子中所做的那样. 于是, 只要解包含回路, 就可以运用尽可能多地增加奇边的运输量并减少偶边的运输量的修改方法, 或者运用相反的策略. 这将产生不包含那个回路且成本更低或有相同成本的解. 如果原来的解包含若干回路, 那么就需要反复使用这一修改策略, 直到断开所有的回路. ◇

根据引理 1, 我们只需要考虑不包含回路的解, 即只考虑那些是生成树或者是生成森林的解; 生成森林是连接所有顶点的树的不相交集合. 我们暂时假设所有无回路解都是生成树. 在下面的讨论中, 我们将解释如何把多条边加入到一个生成森林解 (如果有的话) 来把它转化成生成树解.

求解策略有两个阶段. 首先, 通过特定方法寻找某个初始的生成树解. 其次, 反复修改当前的生成树解来寻找更廉价解, 直到找到一个最廉价的解. 寻找初始解的最简单方法就是使用西北角 (Northwest Corner) 规则. 在练习 8 中, 我们给出获得通常更加接近最优解的初始解的另一种方法.

阶段 I: 使用西北角规则寻找初始解

采用我们的样例运输问题. 实际上, 图 4.19 中的解是利用西北角规则得到的. 这个规则的运用如下:

从运输表的西北角开始, 即, 从项 (1, 1) 开始. 使 x_{11} 尽可能大. 这个值是 s_1 和 d_1 的最小值. 对于我们的问题来说, $x_{11} = \min(40, 60) = 40$. 我们已经用尽了仓库 1 的供应量, 所以从运输表中删除第一行, 并把商店 1 的需求量减少到 $60 - 40 = 20$. 修改后的运输表如图 4.20 所示.

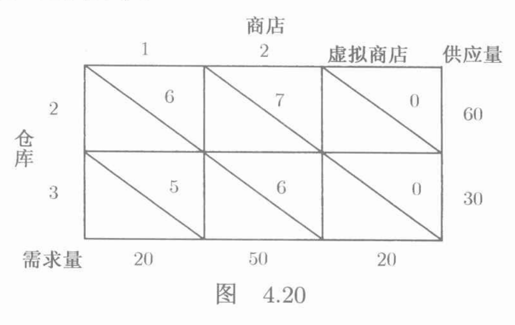


图 4.20



为了满足商店 1 的其余需求, 转到图 4.20 所示的剩余表格的西北角, 即项 (2, 1), 并使该项尽可能大, 所以 $x_{21} = \min(40, 20) = 20$. 现在我们已经满足了商店 1 的需求量, 所以从表格中删除第一列, 并把仓库 2 的供应量减少到 40. 参见图 4.21.

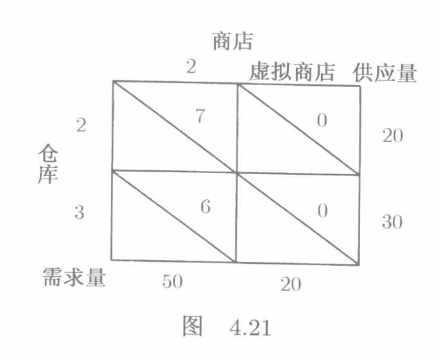


图 4.21

继续这一过程, 使剩余运输表中的西北角的项尽可能大. 完整的西北角规则解见图 4.19, 为了方便起见, 这个解显示在图 4.22 中.

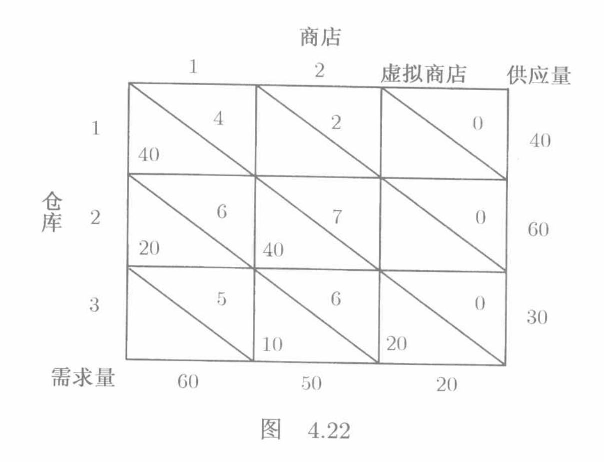


图 4.22

注意, 如果我们分配给当前运输表的当前西北角项的值用尽了第一个 (剩余的) 供应量, 而且还满足第一个剩余商店的需求, 那么必须删除当前运输表的第一行和第一列. 这导致不连通的边集合解, 即, 一个生成森林. 为了回避这一结果, 我们任意地保留第一行或第一列, 尽管它的供应量或需求量是 0.

接下来, 我们解释第二阶段, 如何改进当前解. 确定这一改进需要 3 个步骤.

阶段 II: 寻找更好解

步骤 II.A. 确定仓库和商店的出售价格

此时, 初始解是图 4.22 中的西北角规则解. 现在, 根据初始解中的运输成本, 引入仓库和商店商品的 "出售价格". 为了开始标价过程, 我们需要挑选出仓库 1 的日常用品的任意价格 u_1, 比如说 $u_1 = \$10$. 在初始解中, 我们使用边 (1, 1) 把商品以每单位 $c_{11} = \$4$ 的成本从仓库 1 运输到商店 1. 因此, 如果在仓库 1 以 \$10 的价格买下这一商品, 又把它以 \$4 的成本运到商店 1, 那么在商店 1, 这个商品应该以 $v_1 = \$10 + \$4 = \$14$ 的价格出售.

商店 1 还以每单位 $c_{21} = \$6$ 的成本从仓库 2 进货. 现在, 我们把根据仓库 1 的

价格确定商店 1 的价格的推导反过来. 假设这个商品在商店 1 的价格是\$14, 从仓库 2 运送到商店 1 的成本是\$6, 因此, 商品在仓库 2 的价格应该是 $u_2 = \$14 - \$6 = \$8$. 仓库 2 还以成本 $c_{22} = \$7$ 向商店 2 出货, 所以在商店 2 的价格应该是 $v_2 = \$8 + \$7 = \$15$. 在当前生成树解的边上使用运输成本, 我们可以继续这一过程来计算仓库 3 和商店 3 的出售价格 u_3 和 v_3.

完整的出售价格集是:

$$\begin{array}{ll} \text{仓库}1: u_1 = \$10 & \text{商店}1: v_1 = \$14 \\ \text{仓库}2: u_2 = \$8 & \text{商店}1: v_2 = \$15 \\ \text{仓库}3: u_3 = \$9 & \text{虚拟商店}1: v_3 = \$9 \end{array} \tag{1}$$

概括起来, 设置 u_1 为任意的值. 然后, 如前面例子那样, 对于当前解的每条边 (i, j), 利用条件 $v_j - u_i = c_{ij}$ 确定后继的 u_i 和 v_j. 注意, 利用这种形式的价格计算只有当解所使用的边不形成回路时才是可能的, 对于这一观点的详细讨论参见练习 9.

引理 2 设 S 是使用边集合 $E(S)$ 的运输问题的生成树解, 根据阶段 II.A 的解 S, 设 u_i 和 v_j 分别是仓库 i 和商店 j 的价格. 对 $E(S)$ 中所有边 (i,j) 求和, 这个解的运输成本 $\sum\limits_{i,j} c_{ij}x_{ij}$[对 $E(S)$ 中的边 (i, j) 求和], 等于以各商店的价格销售该商店的需求, 减去在各仓库价格的仓库供应成本所得的利润. 即

$$\sum_{i,j} c_{ij}x_{ij} = \sum_j v_j d_j - \sum_i u_i s_i \tag{2}$$

证明 对于 $E(S)$ 中的每条边 (i, j), 通过条件 $c_{ij} = v_j - u_i$ 确定价格. 因此,

$$c_{ij}x_{ij} = (v_j - u_i)x_{ij} \tag{3}$$

对于 $E(S)$ 中的所有边, 对 (3) 求和, 来自仓库 i 的出货 x_{ij} 的总数是 s_i, 商店 j 的进货的总数是 d_j. 因此, 对于 $E(S)$ 中各边上的 (3) 式, 其和的右边等于 (2) 的右边. 显然, (3) 的这个和的左边等于 (2) 的左边. ◇

为了说明引理 2, 对于有关联价格 (1) 的图 4.22 中的解, (2) 的右边是

$$\begin{aligned} &(\$14 \times 60 + \$15 \times 50 + \$9 \times 20) - (\$10 \times 40 + \$8 \times 60 + \$9 \times 30) \\ =&\$1\ 770 - \$1\ 150 = \$620 \end{aligned}$$

正如前面所提示的那样, \$620 是这个解的运输成本之和.

阶段 II.B. 确定把哪条边加入到当前解中

现在, 我们查看当前解没有使用的边. 考虑边 (1, 2), 它的成本是 $c_{12} = \$2$. 仓库 1 的出售价格是 \$10, 商店 2 的销售价格是 \$15. 然而, 如果在仓库 1 以 \$10 的价

格购买商品, 且在边 (1, 2) 上以 \$2 的价格发货, 则在商店 2 可以以 \$10 + \$2 = \$12 的价格出售, 结果是把商店 2 的每件商品的价格比原来降低 \$3. 商店 2 的价格降低将减少利润, 即商店的销售额减去仓库的购买额, 根据引理 2, 这个利润等于运输成本的减少. 这表明把边 (1, 2) 加入到解中可以得到一个更廉价的解. [当把边 (1, 2) 加入解时, 为了维护生成树解, 应该删除当前解的某条边, 这是在阶段 II.C 中所做的选择.] 在使用边 (1, 2) 之前, 我们检查不在当前解中的其他边, 看一看其中每条边能够减少多少利润. 这些结果概括如下.

$$\begin{aligned} &\text{边}(1,2): c_{12} = \$2 < v_2 - u_1 = \$15 - \$10 = \$5,\ \text{减少}\$3 \\ &\text{边}(1,3): c_{13} = \$0 > v_3 - u_1 = \$9 - \$10 = -\$1,\ \text{增加}\$1 \\ &\text{边}(2,3): c_{23} = \$0 < v_3 - u_2 = \$9 - \$8 = \$1,\ \text{减少}\$1 \\ &\text{边}(3,1): c_{31} = \$5 = v_1 - u_3 = \$14 - \$9 = \$5,\ \text{不变} \end{aligned} \tag{4}$$

我们看到边 (1, 2) 使成本减少最多. 所以, 我们把边 (1, 2) 加入到当前解中. *如果不存在减少解的成本的边, 那么当前解就是最优的, 我们就完成了任务.*

阶段 II.C. 确定更廉价的生成树解

在阶段 II.B 中, 选择一条要加入到当前解的边. 对于图 4.22 的例子, 选择边 (1, 2). 当把这条边加入到当前解的边集合时, 根据练习 3.2 的练习 28 可知, 我们有唯一一条回路. 对图 4.22 的解来说, 这条回路是 (1, 2), (2, 2), (2, 1), (1, 1). 正如在上面引理 1 的证明中所提到的那样, 通过尽可能多地增加奇边 (1, 2) 和 (2, 1) 上的流, 同时相应地减少偶边 (2, 2) 和 (1, 1) 上的流, 我们得到一个更廉价的解. 净减少量是 $c_{12} + c_{21} - c_{11} - c_{22} = 2 + 6 - 4 - 7 = -3$. 注意, 这一计算肯定了我们前面的分析, 即在边 (1, 2) 上, 每单位出货将节省 \$3.

上面做法的极限限制是偶边上的运输量减少到 0. 在我们的例子中, 因为当前出货 x_{11} 和 x_{22} 都是 40, 所以可以把边 (1, 2) 和 (2, 1) 上的运输量增加 40, 而把边 (1, 1) 和 (2, 2) 上的运输量减少 40. 比前面的解廉价 40×\$3=\$120 的新解如图 4.23 所示.

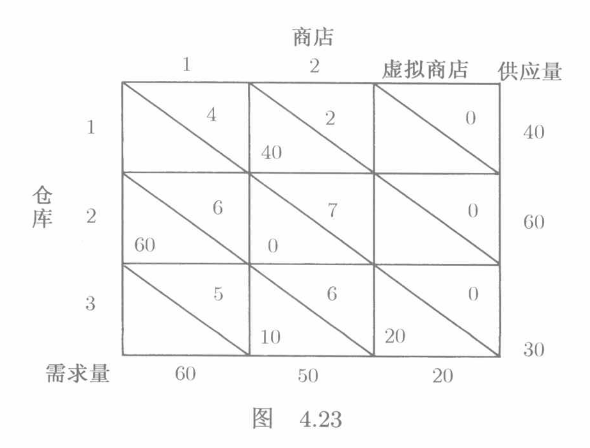


图 4.23

注意, 尽管边 (1, 1) 和 (2, 2) 上的流都减少到 0, 但是我们不能同时删除这两条边. 如果这样做了, 新解将不是生成树. 随意挑出边 (1, 1) 和 (2, 2) 之一, 使其留在新解中, 但是有运输量 0. 在图 4.23 中, 我们挑出 (2, 2).

现在, 我们重复阶段 II 的 3 个部分来寻找更好的解. 在阶段 II.A 中, 使用图 4.23 的解, 我们计算仓库和商店的新价格. 从 $u_1 = \$10$ 和使用边 (1, 2) 开始, 我们找到 $v_2 = u_1 + c_{12} = \$10 + \$2 = \$12$. 如上那样继续下去, 我们得到下面的价格集合.

$$\begin{array}{ll} \text{仓库1}: u_1 = \$10 & \text{商店1}: v_1 = \$11 \\ \text{仓库2}: u_2 = \$5 & \text{商店1}: v_2 = \$12 \\ \text{仓库3}: u_3 = \$6 & \text{虚拟商店}: v_3 = \$6 \end{array} \tag{5}$$

在阶段 II.B 中, 我们查看不在当前解中的边是否产生运输成本上的减少. 计算得到如下结果.

$$\begin{array}{l} \text{边}(1,3): c_{13} = \$0 > v_3 - u_1 = \$6 - \$10 = -\$4,\ \text{增加}\$4 \\ \text{边}(2,2): c_{22} = \$7 = v_2 - u_2 = \$12 - \$5 = \$7,\ \text{不变} \\ \text{边}(2,3): c_{23} = \$0 < v_3 - u_2 = \$6 - \$5 = \$1,\ \text{减少}\$1 \\ \text{边}(3,1): c_{31} = \$5 = v_1 - u_3 = \$14 - \$9 = \$5,\ \text{不变} \end{array} \tag{6}$$

因为没有边使当前解成本减少[①] , 所以当前解是最优的, 我们的工作完成.

练　习　4.5

练习概况　练习 1~7 对运输问题求解. 练习 8 给出了一个更好的开始规则. 练习 9 讨论当解包含回路时不能唯一确定价格的原因.

1. 求解下面的运输问题, 其中仓库 1 有 30 个单位, 仓库 2 有 30 个单位, 而商店 1 需要 20 个单位, 商店 2 需要 40 个单位. 下表给出运输成本.

(a)

商店		1	2
仓库	1	\$6	\$2
	2	\$4	\$3

(b)

商店		1	2
仓库	1	\$4	\$3
	2	\$6	\$4

2. 求解下面的运输问题, 其中仓库 1 有 50 个单位, 仓库 2 有 20 个单位, 而商店 1 需要 20 个单位, 商店 2 需要 30 个单位. 下表给出运输成本. 注意对于超出的供应, 将需要虚拟商店.

(a)

商店		1	2
仓库	1	\$8	\$4
	2	\$3	\$6

(b)

商店		1	2
仓库	1	\$4	\$4
	2	\$5	\$7

---

① 实际上, 按 (6), 把边 (2, 3) 加到边集合可能获得更廉价的解, 因为边 (2, 3) 使当前解的成本减少. 但是, 按阶段 II.C 处理时, 将减少边 (2, 2) 的流到 0, 而它本身就是 0, 因此不产生变化, 只是结果的树含边 (2, 3), 而不含边 (2, 2). 当前解的最优性不变. —— 译者注

3. 求解下面的运输问题, 其中仓库 1 有 30 个单位, 仓库 2 有 30 个单位, 仓库 3 有 30 个单位, 而商店 1 需要 40 个单位, 商店 2 需要 50 个单位. 下表给出运输成本.

(a)

商店		1	2
仓库	1	\$4	\$6
	2	\$5	\$3
	3	\$6	\$8

(b)

商店		1	2
仓库	1	\$5	\$4
	2	\$6	\$5
	3	\$3	\$7

4. 求解下面的运输问题, 其中仓库 1 有 30 个单位, 仓库 2 有 30 个单位, 仓库 3 有 30 个单位, 而商店 1 需要 40 个单位, 商店 2 需要 40 个单位. 下表给出运输成本. 注意对于超出的供应, 将需要虚拟商店.

(a)

商店		1	2
仓库	1	\$5	\$2
	2	\$9	\$5
	3	\$4	\$8

(b)

商店		1	2
仓库	1	\$8	\$3
	2	\$4	\$5
	3	\$4	\$9

5. 求解下面的运输问题, 其中仓库 1 有 30 个单位, 仓库 2 有 30 个单位, 仓库 3 有 30 个单位, 而商店 1 需要 20 个单位, 商店 2 需要 20 个单位, 商店 3 需要 50 个单位. 下表给出运输成本.

(a)

商店		1	2	3
仓库	1	\$4	\$6	\$5
	2	\$2	\$4	\$6
	3	\$5	\$3	\$7

(b)

商店		1	2	3
仓库	1	\$8	\$2	\$4
	2	\$4	\$6	\$4
	3	\$5	\$4	\$5

6. 求解下面的运输问题, 其中仓库 1 有 40 个单位, 仓库 2 有 30 个单位, 仓库 3 有 50 个单位, 而商店 1 需要 50 个单位, 商店 2 需要 10 个单位, 商店 3 需要 40 个单位. 下表给出运输成本. 注意对于超出的供应, 我们需要虚拟商店.

(a)

商店		1	2	3
仓库	1	\$7	\$2	\$5
	2	\$3	\$5	\$4
	3	\$4	\$6	\$3

(b)

商店		1	2	3
仓库	1	\$8	\$3	\$2
	2	\$3	\$5	\$7
	3	\$6	\$4	\$5

7. 求解下面的运输问题, 其中仓库 1 有 30 个单位, 仓库 2 有 30 个单位, 仓库 3 需要 30 个单位, 仓库 4 有 30 个单位, 而商店 1 需要 40 个单位, 商店 2 需要 30 个单位, 商店 3 需要 50 个单位. 下表给出运输成本.

(a)

商店		1	2	3
仓库	1	\$4	\$3	\$6
	2	\$8	\$7	\$4
	3	\$6	\$3	\$7
	4	\$7	\$3	\$8

(b)

商店		1	2	3
仓库	1	\$7	\$5	\$3
	2	\$2	\$4	\$6
	3	\$5	\$6	\$5
	4	\$7	\$3	\$3

8. 这一练习给出了更好的初始生成树解, 称其为*最小成本规则*(Minimum Cost Rule). 在这一方法中, 在每个阶段不是取西北角的项, 而是挑出当前表中的最小成本项. 对于图 4.18 的运输表, 这一最小成本项是 (1, 3)、(2, 3) 和 (3, 3), 这 3 条边的成本都是 \$0. 对于这样的平局, 即同为最小成本的情况, 可以挑出其中的任意一项. 假设我们挑出项 (1, 3). 最小成本规则尽可能多地沿着边 (1, 3) 运输, 即, $x_{13} = \min(60, 20) = 20$. 现在删除虚拟商店这一列, 因为它的需求已经得到满足, 并把仓库 1 的供应减少到 $60 - 20 = 40$. 在简化后的运输表中, 最小成本项是 (1, 2), 其成本是 \$2. 所以我们尽可能

多地沿 (1, 2) 运输. 于是, $x_{12} = \min(40, 50) = 40$. 现在删除仓库 1 这一行, 因为它的供应已经用完, 而且把商店 2 的需求量减少到 $50 - 40 = 10$. 我们按着这一模式继续.

使用最小成本规则求解下面的运输问题:

(a) 练习 4(a); **(b)** 练习 5(a); **(c)** 练习 6(a).

9. 对图 4.18 的运输表运用阶段 II.A, 尝试着确定仓库和商店的出售价格. 因为这一运输表中的解使用形成一条回路的边, 证明存在两条确定商店 3 的销售价格的途径. 由这两个不同的途径确定的商店 3 的销售价格相同吗?

4.6 小结及参考文献

本章对 3 个基本网络优化问题给出了算法: 最短路径、最小生成树、最大流以及运输问题. 我们把重点放在了最大流的讨论上. 我们说明了如何把这些流用于其他各种网络问题. 4.4 节使用流模型探讨了匹配的组合理论. 本章所有关于流的内容在 Ford 和 Fulkerson 的开创性著作《网络流》(见文献 [2]) 和 Ahuja 等人的《网络流理论》(见文献 [1]) 中都有更详细的讨论. 我们省略了关于算法速度的讨论. 有兴趣的读者可以参考 Ahuja 等人的著作 [1], 该书讨论了最短路径算法和最大流算法的有效实现. n 顶点网络的现代最大流算法需要少于 $O(n^3)$ 的操作.

在网络流问题中, 我们自然地把成本与边相关联, 使得当存在很多可能的最大流时, 你可以寻找最小成本最大流. 这样的问题称为**运输出货**(trans-shipment) 或**运输**(transportation) 问题. 类似地, 在有很多解 (X 匹配) 的匹配问题中, 你可以寻找最小成本匹配. 这样的问题称为**分配**(assignment) 问题. 这些最小化问题都存在高效算法 (参见 [1]、[2] 或 [3]). 另外, 对于任意的流优化问题, 无论它是否包含上面所提到的最小化, 它都是边流的线性函数的优化问题, 这些边流满足特定的线性等式或线性不等式, 诸如 4.3 节的流限制 (a)、(b) 和 (c). 这样的受限线性优化问题称为**线性规划**(linear program). 线性规划是运筹学的核心工具, 而且存在对线性规划求解的好算法. 然而, 本章所提到的面向网络的算法可以更加有效地解决网络问题.

1. R. Ahuja, T. Magnanti, and J. Orlin, *Network Flow Theory, Algorithms and Applications,* Prentice-Hall, Englewood Cliffs, NJ, 1993.

2. L. Ford and D. Fulkerson, *Flows in Networks,* Princeton University Press, Princeton, NJ, 1962.

3. F. Hillier and G. Lieberman, *Introduction to Operations Research,* Holden Day, San Francisco, 1988.

4. B. Schwartz, "Possible winners in partially completed tournaments," *SIAM Review* 8(1966), 302–308.

第二部分　枚　　举

第 5 章　排列和选择的一般计数方法

5.1　两个基本计数法则

本章是本书中最重要的一章. 它开发组合推理的基础技巧. 我们给出几个基本公式, 其中包括排列和组合公式, 其中大部分公式读者在前面已经看过. 然后, 我们研究包含计数的各种字问题, 并说明如何把它们分解成简单数值因子的和与积. 在示例学习之后, 接下来应该积极应对练习. 每一节最后的前半部分练习类似于该节讨论过的例子. 而后半部分练习尽管也要求相同的一般逻辑推理, 但是它们与例子有很大的不同, 需要敏锐的洞察力和数学建模能力. 对于大多数组合应用, 熟练运用这 3 个基本的问题求解技巧, 并辅以不断发明的专门技术, 是成功的关键. 这些技巧在构建计算机程序和掌握很多计算机科学的理论课题中非常有帮助. 总之, 对于大部分学生来说, 本章是本书中最具挑战且最有价值的一章.

本书关于枚举的其余章节研发专门的理论和技术, 这些理论和技术很大程度上简化了特殊类型计数问题的解决方案. 在你只使用前面几个法则设法解决了本章的计数问题之后, 这些理论的用途就会相当明了. 遗憾的是, 不存在对现实世界的众多计算问题的解决方案都有帮助的理论.

为了激发我们解决问题的兴趣, 我们给出概率和统计、计算科学、运筹学以及其他学科的应用. 然而, 如果你不去积极涉入这一应用领域, 那么这些应用中的计数问题的细节通常显得单调而乏味. 所以我们将以已经明了的关于娱乐的例子, 诸如纸牌概率等例子 (和练习) 为出发点. 这些问题的解决方案需要大多数现实应用中使用的数学技术.

本节从两个初等但却又很基础的计数法则开始, 这两个法则的简单性既掩盖了它们的威力, 又让人容易误用.

加法原理　如果第一个集合有 r_1 个不同对象, 第二个集合有 r_2 个不同对象, $\cdots\cdots$, 第 m 个集合有 r_m 个不同对象, 而且如果这些集合互不相交, 那么从这 m 个集合中选出一个对象的方法数是 $r_1+r_2+\cdots+r_m$.

乘法原理　假设一个过程可以分解成 m 个连续 (有序的) 阶段, 第一个阶段有 r_1 个不同的结果, 而第二个阶段有 r_2 个结果, $\cdots\cdots$, 第 m 个阶段有 r_m 个结果. 如果每一个阶段的结果数与前面各阶段的选择无关, 而且如果复合结果都是互不相同的, 那么整个过程有 $r_1\times r_2\times\cdots\times r_m$ 个不同的复合结果.

记住, 加法法则要求对象集合互不相交, 而乘法法则要求过程分解成有序的阶段, 而且复合结果互不相同. 这两个法则的正确性可以直接由整数乘法和加法的定义得到. 即, 当把 a 个项的集合加入到有 b 个项的集合时, 和 $a+b$ 就是结果的项

的数量; 积 $a \times b$ 是序列 (A, B) 的数量, 其中 A 是 a 项中的任意项, B 是 b 项中的任意项. 这两个法则是二元运算的标准的 m 阶扩展.

例 1　增加一组学生

Mindthumper 教授的代数班有 40 名学生, 几何班有 40 名学生. 这两个班一共有多少名不同的学生?

倘若没有学生同处于两个班, 那么根据加法法则, 答案是 80 名学生. 假设有 10 个学生同处两个班. 为了得到学生的不相交集合, 我们把学生分为只在代数班、只在几何班和同时在两个班这 3 类. 因为有 10 个学生同处两个班, 所以有 $40 - 10 = 30$ 名学生只在代数班. 类似地, 有 30 名学生只在几何班. 现在可以安全地使用加法法则对 3 个互不相交集合中的学生数量求和, 总的学生数量是 30+30+10=70. □

例 2　投掷骰子

要投掷两枚骰子, 一枚是绿的, 一枚是红的.

(a) 这个过程有多少种不同的结果?

(b) 不成双的结果 (两枚骰子上没有相同值) 的概率是多少?

(a) 单枚骰子有 6 种结果, 所以根据乘法法则, 这个过程总共有6×6=36种结果.

(b) 我们使用某一事件的概率公式来计算没有相同值的结果的概率: 产生理想事件的结果子集的大小除以所有可能结果 (其细节参见附录 A.3). 根据部分 (a) 的结果, 概率分数的分母是 36. 为了得到分子, 即没有相同值的结果数量, 我们使用乘法法则. 假设投掷红色骰子是这一过程的第一阶段. 第一阶段中有 6 种可能的结果. 一旦投出红色的骰子, 那么对于绿色骰子来说, 就有 5 种允许值, 而且它们与红色骰子的特定值无关. 所以有 6×5=30 种没有相同值的结果, 所以两枚骰子没有相同值的概率是 30/36=5/6. □

注意, 为了可以在例 2 的第二部分使用乘法规则, 我们必须把 “没有相同值” 的限制重述为 “绿色骰子上的值不同于红色骰子上的值”. 寻找一种限制的适当重述, 使得我们可以使用加法和乘法法则, 是求解组合问题的具有挑战性的 “艺术”. 在例 2(b) 中, 没有相同值的结果数量还可以通过从所有可能结果中减去有相同值的 6 种结果而得出, 这时产生 $36 - 6 = 30$ 种没有相同值的结果.

例 3　排列书

有 5 本不同的西班牙语书, 6 本不同的法语书, 8 本不同的特兰西瓦尼亚语书. 有多少种方法可以取两本不相同语言的书?

如果选取一本西班牙语书和一本法语书, 那么由乘法法则, 可以有 5×6=30 种方法做这种选择; 如果选择一本西班牙语书和一本特兰西瓦尼亚语书, 那么有 5×8=40 种方法, 如果选取一本法语书和一本特兰西瓦尼亚语书, 那么有 6×8=48 种方法. 这 3 种类型的选择是互不相交的, 所以根据加法法则, 总共有 30+40+48=118 种方法. □

前面的例子是组合问题求解中的基本思维方式的代表: 首先, 总是尝试把一个

问题分解成数量适中的易于处理的子问题. 可能存在解决这一问题的更加聪明的方法, 但是如果能够把原始问题化简成我们比较熟悉的一组子问题, 那么犯错的可能性就会减少.

例 4 字母序列

有多少种方法使用字母 a, b, c, d, e, f 构成三字母序列? **(a)**允许有重复的字母? **(b)**任何字母都不得重复? **(c)**没有重复且包含字母 e? **(d)**可以有重复且包含字母 e?

(a)有重复, 对于这个序列的每一后继字母我们有 6 种选择. 所以根据乘法法则, 有 $6\times6\times6=216$ 种可以有重复的三字母序列.

(b)没有重复, 对于第一字母, 我们有 6 种选择. 对应于剩余的 5 个字母 (无论第一个字母选择什么), 第二个字母有 5 种选择. 类似地, 对于第三个字母, 有 4 种选择. 所以有 $6\times5\times4=120$ 种没有重复的三字母序列.

(c)往往做一个图来展示序列中的位置有帮助, 即使这个序列只有 3 个字母:

$$_\ _\ _$$

这样的图帮助我们集中考虑所要选择的位置. 如果序列必须包含 e, 那么对于序列中 e 的位置有 3 种选择, 如下面的 3 个图所示:

$$\underline{e}\ _\ _ \qquad _\ \underline{e}\ _ \qquad _\ _\ \underline{e}$$

在每一个图中, (e 之外的) 剩余 5 个字母进入剩余两个位置的第一个位置, 有 5 种选择, 剩余 4 个字母进入另一个位置, 有 4 种选择. 因此, 总共有 $3\times5\times4=60$ 种包含 e 的三字母序列. 练习 19 给出另一种解决这一问题的方法.

(d)当允许有重复时, 尝试部分 (c) 使用的方法. 如前所述, 对于 e 的位置有 3 种选择. 对于这些选择中的任意一种选择, 其他两个位置总共有 $6\times6=36$ 种选择, 因为 e 和另一字母可以不只出现一次. 但是答案 $3\times36=108$ 是不正确的.

我们违反了乘法法则, 因为这些结果不是互不相同的. 考虑这样的序列

$$\underline{e}\ \underline{c}\ \underline{e}$$

在上面的计算过程中, 我们两次生成这个序列: 一次是当 e 被放置在第一个位置时, 随后是 c e, 对于后面两个位置来说, 这是 36 种选择中的一种; 第二次是当 e 被放置在最后一个位置而 e c 放置在前两个位置.

必须使用一种方法把原始问题分解成确保结果互不相同的部分. 我们根据第一个 e 在序列中出现的位置, 把问题分解成不相交的情况. 首先假设第一个 e 在第一个位置上:

$$\underline{e}\ _\ _$$

第二个位置和第三个位置各有 6 种选择 (包括 e), 共有 $6\times6=36$ 种方法.

接下来, 假设第一个 e 在第二个位置上:

$$\underset{\text{非e}}{\underline{\quad}}\ \overset{\text{e}}{\underline{\quad}}\ \underline{\quad}$$

那么, 第一个位置有 5 种选择 (不能是 e), 最后一个位置有 6 种选择, 总共有 5×6 种方法. 最后, 让第一个 (唯一的)e 在最后位置上:

$$\underset{\text{非e}}{\underline{\quad}}\ \underset{\text{非e}}{\underline{\quad}}\ \overset{\text{e}}{\underline{\quad}}$$

前两个位置各有 5 种选择, 共有 5×5 种方法. 因此正确答案是 (6×6)+(5×6)+(5×5)=91 种选择.

□

解决大多数计数问题的最困难部分是在问题中寻找这样一种结构, 它能够把原始问题分解成子情况或子阶段. 换句话说, 困难之处是 "开始". 与此同时, 必须确保这种分解能产生结果完全互不相同的子情况或阶段.

很多计数问题都需要考虑它们各自的特点. 对于这样的问题, 知道问题 A 的解决方法对求解问题 B 没有什么帮助. 学习如何构建这类计数问题的解决方案是组合数学课程的内容. 只通过阅读课本中的例子是无法得到这样的技巧的. 必须通过自己实际地去解决大量的问题.

下面的例子说明在求解组合问题中这种特殊的观察.

例 5 非空组合

由 5 个 (不可区分的) 苹果, 8 个 (不可区分的) 橙子可以形成多少不同的非空组合?

有解决组合问题经验的读者也许会想到, 根据组合中对象数目把这一问题分解成若干子情况. 而其中任意一种子情况都可以相当容易地计算出来. 但是一共有 13 种子情况: 这一组合可能有 1, 2, $\cdots$ 一直到 13 个水果.

在计数不同的可能性的时候, 我们必须注意什么使一种组合与另一种组合不同. 这一问题的答案是苹果数量或橙子数量在不同组合中是不同的. 于是可以通过一个整数对 (a, o) 刻画任意一种组合, 其中 a 是苹果数量, o 是橙子数量.

现在就很容易计数组合数了. 对于 a 来说共有 6 种可能的值: 0, 1, 2, 3, 4, 5, 而对于 o 则有 9 种可能值. 所以一共有 6×9=54 种不同的组合. (注意我们用 6 乘以 9, 而不是把它们加起来, 因为一个组合是把任意数量的苹果与任意数量的橙子放在一起; 如果想要计数若干个苹果或若干个橙子, 而不是二者总量的各种情况时, 则要把 6 和 9 加起来.)

因为要求的是非空组合, 前面允许的可能情况之一是 (0, 0), 所以答案是 54−1 =53.

□

当你遇到障碍而对某个问题无从下手时, 这里有一条可以考虑的忠告. 尝试着系统地把你想要枚举的一些可能结果 (例如十几个结果) 写下来. 在列出结果的过程中, 你应该留意某种模式的出现. 把你的列表想象成某种特殊子情况的一部分. 然后问自己, 为了完成这种子情况, 你的列表还需要包含多少结果? 接下来, 问自己

还有其他什么样的子情况需要计数. 一旦把第一种子情况成功地分析出来, 那么其他子情况通常更加容易.

下面概括在求解组合问题中已经遇到的两个重要方面. 第一, 必须寻找重述问题中的限制的方法, 使得我们可以把加法和乘法法则组合起来加以运用. 第二, 为了使用这些法则, 必须把问题分解成各种情况或各个阶段, 并保证不同情况的结果互不相同.

练 习 5.1

练习概况 前面的练习比较直接. 后面的练习更具挑战性, 每一个问题需要不同的分析. 对于较难的问题, 读者必须研究这些问题特有的解决方法, 而不是模仿曾经在正文的某个例子中使用过的方法. 所有练习都应该仔细阅读两遍, 以防曲解.

某些问题给出可能的答案以供参考. 尝试着根据每一个表达式推断出应该产生这样的答案的推理. 如果答案是错误的, 指出推理中的错误.

词语 "之间" 总是意指包含的意义; 即, "0 与 50 之间的整数" 意指 0, 1, 2, $\cdots$, 49, 50. 特殊结果的概率是这一特殊结果的数量除以总结果数. 关于概率的更多内容, 参见附录 A.3.

1. **(a)** 有多少种方法从 HOUSE 中选出两个不同字母的序列? 对于 MATHEMATICS 又如何?

(b) 有多少种方法从 HOUSE 中选出第一个字母是元音字母, 其后是一个辅音字母的序列? 对于 MATHEMATICS 又如何?

2. **(a)** 0 到 40 之间 (包含两端) 有多少个整数?

(b) 这些整数中, 可以被 2 整除的整数有多少个?

(c) 这些整数中, 差为 5 的 (无序) 整数对有多少?

3. 一家商店运来 7 种款式的裤子. 每一种款式都有 10 种不同可能的腰围大小、6 种裤长和 4 种颜色. 这家商店有多少种不同类型的裤子?

4. 如果把一枚硬币投掷 100 次, 有多少种不同的正反面序列? 利用 $2^{10}=1\,024\approx1\,000=10^3$, 用 10 的幂给出近似答案.

5. 有多少五字母 "单词"(可以重复的任意五字母序列)? 没有重复的五字母单词有多少?

6. 从一组 n 对已婚夫妇中挑出不是夫妻的一个男人和女人有多少种方法?

7. 给定 9 本不同的英语书, 7 本不同的法语书和 5 本不同的德语书:

(a) 挑出 1 本书有多少种方法?

(b) 挑出 3 本书, 每种语言各 1 本, 有多少种方法?

(c) 把 3 本书摆成一行, 其中正好缺 1 种语言 (这 3 本书的顺序产生差异), 有多少种方法?

8. 城市 A 和 B 之间有 4 条不同的道路, B 和 C 之间有 3 条不同的道路, A 和 C 之间有 2 条不同的道路.

(a) 从城市 A 到城市 C 总共有多少条路线?

(b) 从城市 A 到城市 C 且返回到城市 A 有多少条路线 (在每个方向上, 任何道路可以使用一次)?

(c) 问题 (b) 的路线中, 从城市 A 到城市 C 并返回的路线中至少经过 B 一次的路线有多少条?

(d) 问题 (b) 的路线中, 从城市 A 到城市 C 并返回的路线中不使用任何道路两次的路线有多少条?

9. 从标准的 52 张纸牌中选出两张不同的纸牌, 使得得到下面的结果, 有多少种方法?

(a) 第一张纸牌是 A, 第二张不是 Q.

(b) 第一张纸牌是黑桃, 而第二张不是 Q.(提示: 注意黑桃 Q.)

10. 3 个 A 和 5 个 B 可以形成多少个非空的字母组合?

11. 有多少方法投掷两个不同的骰子, 使得生成的点数和能被 3 整除?

12. 有多少个四字母 "单词"(可以重复的字母序列), 它们的第一个字母和最后一个字母都是元音? 有多少 "单词", 其元音 (如果出现的话) 只能出现在第一个位置和最后一个位置?

对上面第二问的可能答案做出评价: **(a)** $5^2 21^2$, **(b)** $5^2 26^2$, **(c)** $21^2 26^2$, **(d)** $26^2 - 21^2 26^6$.

13. (a) 有多少个不同的 5 位数 (不允许如 00 174 这样以 0 开头的数)?

(b) 有多少个偶数值 5 位数?

(c) 有多少个正好有一个 3 的 5 位数?

(d) 有多少个 5 位回文数 (当把一个数的各位的顺序反过来时得到相同的数, 例如, 15251)?

14. 6 个数字 1, 1, 1, 1, 2, 3 的不同排列可以形成多少不同的数?

15. 已经洗过的一副纸牌的头两张牌不是一对的概率是多少?

16. (a) 当连续投掷一个红色一个白色的一对骰子两次时, 有多少不同的可能结果?

(b) 第二次投掷与第一次投掷, 每一枚骰子显示相同值的概率是多少?

(c) 两次投掷, 两枚骰子的点数和相同的概率是多少?

(d) 第二次投掷时两枚骰子的点数和较大的概率是多少?

17. 让一个人给另一个人打电话, 这个人再给另一个人打电话, 以此类推连续地打下去, 一条传闻在 10 个人之间随机传播. 一个人可以把传闻告诉给除刚刚给他打电话的人之外的任何一个人.

(a) 通过 3 次电话在这组人中传递, 传闻可以有多少不同的路径? 通过 n 次通话呢?

(b) 如果 A 开始传播传闻, 那么 A 接受第 3 次电话, 这样的概率是多少?

(c) 如果 A 不开始传播传闻, 那么 A 接受第 3 次电话, 这样的概率是多少?

18. (a) 设汽车牌照由 3 个字母和 3 位数字组成, 如果这 3 个字母必须在一起或者出现在开头或者出现在结尾, 那么有多少不同的牌照?

(b) 设汽车牌照分别包含 1 个字母、2 个字母或 3 个字母和 1 位数字、2 位数字或者 3 位数字, 则各有多少不同的牌照? 假设字母必须在一起.

19. 首先计数没有 e 的三字母序列数来重新求解例 4 的问题, 即计数使用字母 a, b, c, d, e, f 且只包含一个 e 的无重复三字母序列数.

20. 在 20 和 40 之间随机选择的两个整数之和是偶数的概率是多少 (允许这两个整数相等)?

评价下面的答案: **(a)** 1/2, **(b)** 11/21, **(c)** $11^2/21^2$.

21. 使用字母 a, b, c, d, e, f 可以形成多少或者包含 e 或者包含 f(或者包含二者) 的不重复三字母序列?

评价下面的答案: **(a)** $3\times2\times4\times3$, **(b)** $3\times2\times5\times4$, **(c)** $3\times2\times4\times4-3\times2\times4$, **(d)** $6\times5\times4-4\times3\times2$.

22. 由 0, 1, 2 构成的长度为 10 的三进制序列中, 相继数对各不相同的序列有多少?

23. 在 1 000 与 10 000 之间有多少满足下面条件的整数 (确保避免 0 开头数字序列):

(a) 没有重复数字?

(b) 允许数字重复, 但没有 2 和 4?

(c) 没有重复数字, 且 2 和 4 至少出现一个?

24. 有多少至少以两个 3 开头并结尾的 12 位十进制序列?

25. 使用数字 0, 1, 2, $\cdots$, 9 能够形成多少长度为 5, 且 10 个数字中正好有两个数字出现的序列, 如 05550?

26. 0 到 9 999 之间的整数中正好有一个 8 和一个 9 的整数的概率是多少?
评价下面的答案: (a)$4\times3/10^4$, (b)$4\times3\times8\times7/10^4$, (c)$2\times4\times3\times8^2/10^4$.

27. 使用字母 A, B, C, D 可以形成多少可重复且不包含单词 BAD 的序列? 例如, 序列 A<u>BAD</u>D 被排除在外.

28. (a) 20 个人投票选举, 共有 7 个候选人, 每人只选一个候选人, 有多少可能的选举结果 (这一结果不仅包含总数, 而且还包含谁投了哪一位候选人的票)?

(b) 如果只有 1 个人投候选人 A 的票, 且只有 1 个人投候选人 D 的票, 那么有多少可能的选举结果?

29. 有 15 个不同的苹果和 10 个不同的梨. Jack 选出一个苹果或一个梨, 然后 Jill 挑出一个苹果和一个梨的可能方法有多少?

30. 当列出 1 到 100 000 之间的所有数时, 数字 5 的书写次数是多少?
评价下面的答案: (a) 4, (b) 5×10^4, (c) 1+10+100+1 000.

31. 当列出 1 到 100 000 之间的所有数时, "25" 被书写多少次? (这个练习是前面练习的扩展.)

32. 如果从单词 RECURRENCE 中随机选出一个字母, 再从 RELATION 中随机选取一个字母, 这两个字母一样的概率是多少?

33. 由数字 1, 2, 3, 4, 5 可以形成多少能被 4 整除的四位数 (可以重复)?

34. 由 n 个 A、n 个 B、n 个 C 和 n 个 D 可以形成多少非空字母组合?

35. 有标号从 1 到 50 的 50 张卡片. 随机选出两张不同的卡片. 一个数是另一个数的两倍的概率是多少?

36. 如果在 1 到 100 之间随机选出两个不同的数, 那么这两个数的差是 15 的概率是多少?

37. 如果投掷 3 枚不同的骰子, 最大值是最小值的两倍的概率是多少?

38. 由数字 3, 4, 4, 5, 5, 6, 7, 7, 7 中的两个或更多数的积可以形成多少个不同的数?

39. 一次聚会上有 10 个人. 有多少种方法可以把他们两两配对形成一个 5 对组合?

40. 在新年的第 1 周把一封连锁信寄给 5 个人. 第 2 周接到信的每一个人再给新的 5 个人寄这封信, 以此类推. 前 5 周有多少人接到了信?

41. 把两个不可区分的车放置在一个 8×8 的棋盘上的同一行或同一列上有多少种方法? 对于 $n \times m$ 的棋盘又如何?

42. 把两个不可区分的王放置在一个 8×8 的棋盘上, 使得这两个王不在相邻的方格内, 有多少种方法? 对于 $n \times m$ 的棋盘情况又如何呢?

43. 把两个不可区分的后放置在一个 8×8 的棋盘上, 使得它们不在同一行、同一列或同一对角线上, 有多少种方法?

44. 1, 3, 5, 10, 20, 50, 82 中的两个或更多数的和可以形成多少个不同的正整数?

45. 一个人有多少方法邀请他的 10 个朋友的非空子集中的人参加晚宴?

46. 在一个 8×8 的棋盘上, 可以画出多少个不同矩形 (矩形的边长是 1 到 8, 如果两个矩形包含不同的方格子集, 那么它们不相同)?

47. 把一枚红色西洋跳棋棋子和一枚黑色棋子放置在一个棋盘的两个黑色格子上, 使得红色棋子可以跳过黑色棋子, 有多少种方法? (棋子从前向后在对角线上跳动.)

48. 使用归纳法给出下面结果的形式证明:

(a) 加法法则; **(b)** 乘法法则.

49. 在实数轴上, 把 n 个白钉子放在位置 $1, 2, \cdots, n$ 上, 把 n 个蓝钉子放置在位置 $-1,-2,\cdots,-n$ 上 (位置 0 空着). 白钉子只能向左移动, 蓝钉子只能向右移动. 当旁边有一个空位置时, 钉子可以移动一个单位占据该位置 (又要它在要求的方向上). 如果一种颜色的钉子在一个其后是空位置的另一种颜色的钉子的前面 (在要求的方向上), 那么这个钉子可以跳过两个单位占据那个空位置 (被跳过的钉子不移动). 通过这两种类型的移动序列 (不必是白钉和蓝钉交替移动), 你可以交换白钉和蓝钉的位置. (参见 *Mathematics Teacher*1982 年 1 月刊中关于这一游戏的文章.)

(a) 对于 $n=3$ 和 $n=4$, 玩这一游戏.

(b) 使用组合观点证明: 对于一般情况, 需要 n^2+2n(单位步和跳跃) 次移动才能完成这一游戏.

5.2 简单排列和选取

n 个不同对象的**置换**(permutation) 是这 n 个对象的排列, 或者排序. n 个不同对象的 ***r* 置换**(r-permutation) 是使用 n 个对象中的 r 个对象的排列. n 个不同对象的 ***r* 组合**(r-combination) 是 n 个对象中的 r 个对象的无序选取, 或子集. 我们用 $P(n,r)$ 和 $C(n,r)$ 分别表示 r 置换的数目和 r 组合的数目. 根据乘法法则, 我们得到:

$$P(n,2) = n(n-1), \qquad P(n,3) = n(n-1)(n-2)$$
$$P(n,n) = n(n-1)(n-2) \times \cdots \times 3 \times 2 \times 1$$

在枚举 n 个对象的所有置换时, 排列的第一个位置有 n 种选择, 第二个位置有 $n-1$ 种选择 ($n-1$ 个剩余的对象),$\cdots$, 最后一个位置有 1 种选择. 使用记号 $n! = n(n-1)(n-2) \times \cdots \times 3 \times 2 \times 1$($n!$ 读作 "n 的阶乘"), 于是有公式

$$P(n,n) = n!$$
$$P(n,r) = n(n-1)(n-2) \times \cdots \times [n-(r-1)] = n!/(n-r)!$$

可以使用 $P(n,r)$ 的公式得到 $C(n,r)$ 的公式. n 个对象的所有 r 置换可以这样生成: 首先选出 n 个对象的 r 组合, 然后以任意顺序排列这 r 个对象. 因此 $P(n,r) = C(n,r) \times P(r,r)$, 求解 $C(n,r)$, 有

$$C(n,r) = \frac{P(n,r)}{P(r,r)} = \frac{n!/(n-r)!}{r!} = \frac{n!}{r!(n-r)!}$$

数值 $C(n,r)$ 因其在二项式 $(x+y)^n$ 的展开式中所扮演的角色, 通常称为**二项式系数**. 我们将在 5.5 节研究与二项式系数有关的二项展开式及恒等式. 通常, 把表达式 $C(n,r)$ 写成 $\binom{n}{r}$, 并说成 "n 选 r"[本书通常写成 $C(n,r)$]. 注意, $C(n,r)=C(n,n-r)$, 因为从 n 个不同对象中挑出有 r 个对象的子集的方法数等于扔掉有 $n-r$ 个对象的子集的方法数.

本节其余内容是继续研究前一节所讨论的求解组合问题的例子. 由于置换和组合的引入, 这些例子包含更复杂的情况. 由于要遵守很多新限制, 所以读者可能会发现自己有些糊涂了, 甚至对于一些基本的问题, 例如什么时候做加法, 什么时候做乘法都会感到有些困惑. 在计数问题的分析中, 每一步都必须小心.

例 1 给魔法师定级

有多少种方法给 n 名大魔法师 (chief wizard) 定级呢? 如果这种定级是随机的 (每一种定级的机率相等), 第 5 名候选人 Gardalf 排在第二个位置的概率是多少?

一个定级就是 n 名候选人的一个排列或置换. 所以有 $n!$ 种定级. 直观地说, 随机定级应该随机安排 Gandalf 的位置, 所以我们希望他有 $1/n$ 的概率在第二个位置上, 或被安排在其他任意指定的位置上. 同理, n 名候选人中的每名候选人在第二个位置的概率是相等的. 一般地, 我们利用一个事件的概率计算 Gandalf 在第二个位置的概率: 产生期望事件的结果子集除以所有可能的结果 (详见附录 A.3).

P(Gandalf 在第二个位置)= Gandalf 在第二个位置的定级数量/所有定级数量

首先, 把 Gandalf 安排在第二个位置上, 有 1 种方法, 然后, 把剩余的候选人安排在其余 $n-1$ 个位置, 有 $(n-1)!$ 种方法. 所以 Gandalf 排在第二个位置有 $(n-1)!$ 种定级, 所以 P(Gandalf 在第二个位置)$=(n-1)!/n!=1/n$. □

从这个例子中, 我们吸取两条重要的经验.

(1) 存在两种分析排列问题的方法. 我们有两种选择: 或者选出哪一项进入第一个位置, 然后哪一项进入第二个位置, 等等; 或者为第一个项挑出哪一个位置, 为第二个项挑出哪一个位置, 等等; 在 Gandalf 排第二位置的问题中, 我们的做法是为项挑选位置.

(2) 为了计数固定了某个特殊位置的排列, 例如在前面的例子中 Gandalf 在第二个位置上, 我们不是计数使 Gandalf 在第二个位置的方法 (只有一种方法把 Gandalf 安排在第二个位置), 而是计数在其余位置上安排其余候选人的方法. 我们计数不受限制的对象.

例 2 可重复字母排列

有多少种方法排列单词 SYSTEMS 中的 7 个字母? 3 个 S 连续出现的排列有多少个?

通过选出 4 个字母 E, M, T, Y 在 7 字母排列中所占据的位置, 然后 3 个 S 将以一种方式填充剩余的 3 个位置, 这样就可以避开多个 S 带来的困难. E 有 7 种可能位置, M 有 6 种可能位置, T 有 5 种可能位置, Y 有 4 种可能位置. 因此有

$P(7,4) = 7!/3! = 840$ 个排列. (计数有重复对象的排列的一般公式将在 5.3 节给出.)

同样, 在这里用连续项应该放置在什么地方, 而不是什么应该进入连续位置来计数排列.

下面考虑 3 个 S 连续出现的情况, 即, 3 个 S 是并肩在一起. 处理这种新的连续性限制的 "技巧" 是, 要认识到当把 3 个 S 捆在一起放置时, 它们现在变成了一个合成字母. 所以这个问题就简化成排列 5 个不同的字母, Y, T, E, M, SSS(把它看成单一字母), 完成这一排列有 5! =120 种方法.

研究这一问题的另一种方法是, 暂时把其中的两个 S 放到一边, 考虑 5 个剩余字母 Y, T, E, M, S 的排列, 有 5! 种方法, 然后在所产生的每一种排列中, 把其余的两个 S 插入到 S 的旁边. □

例 3 二进制序列

6 个 1 和 2 个 0 可以形成多少不同的 8 位二进制序列?

一个有 6 个 1 的 8 位二进制序列与另一个这样的序列的不同点是什么? 显然答案是这 6 个 1 的位置 (或者 2 个 0 的位置). 尽管最初这个问题看起来像是一个排列问题, 但是必须计数的是 6 个 1 所有不同可能的放置, 即, 在这个二进制序列中, 8 个位置中的 6 个位置的不同可能子集. (在位置 1, 2, 3, 4, 5, 6 上有 1 的序列与在位置 6, 5, 2, 3, 4, 1 上有 1 的序列相同, 顺序不重要, 我们只关心相关位置的集合.) 所以, 答案是 $C(8, 6)=28$.

也可以从 8 个位置中挑出两个位置来放置 0. □

例 4 纸牌概率

(a) 标准的 52 张纸牌可以形成多少手 5 张牌 (子集)?

(b) 如果随机选出一手 5 张牌, 那么得到同花 (手中的 5 张牌都是同一花色的牌) 的概率是多少?

(c) 得到 3 张 A 但不是 4 张 A 的概率是多少?

(a)一手 5 张牌是从一副 52 张纸牌中选出的 5 张牌的子集, 所以有 $C(52,5) = 52!/(47!5!) = 2\,598\,960$ 手不同的 5 张牌.

(b)为了求同花的概率, 需要求所有牌有相同花色的 5 张牌子集的数量. 一共有 4 种花色, 在给定的花色中从 13 张牌中取出有 5 张牌的子集的方法有 $C(13,5) = 13!/(5!8!) = 1\,287$ 种. 所以共有 $4\times 1\,287=5\,148$ 种同花, 所以有

$$\text{P(手中的 5 张牌都是同一花色)} = \frac{5\,148}{2\,598\,960} = 0.001\,98(\approx 0.2\%)$$

(c)为了计数正好有 3 张 A 的手牌数, 必须从 4 个 A 中选取 3 个, 共有 $C(4, 3)= 4$ 种方法, 然后, 从 48 张非 A 牌中取 2 张牌填充这手牌的其余两张牌, 共有 $C(48, 2)=1\,128$ 种方法. 所以总共有 $4\times 1\,128=4\,512$ 手刚好有 3 张 A 的牌, 所以有:

$$P(\text{手中的 5 张牌刚好有 3 张 A}) = \frac{4\ 512}{2\ 598\ 960} = 0.001\ 74$$ □

注意, 为了计算例 4 中有 3 张 A 的手牌数, 我们隐式地使用了乘法法则来把选取 3 张 A 的方法与填充 2 张非 A 牌的方法相乘. 然而, 一手牌是一个无序集合, 通过把它排序成两个部分, 我们可能生成了实际上是相同集合的两个结果, 这违背了乘法法则中结果互不相同的条件. 在这一问题中, 可以安全地把一手牌分解成 A 部分和非 A 部分, 因为两部分中的纸牌数量不同. 下面的例子说明违背了这一互不相同条件的情况.

例 5　组建委员会

要从有 7 名女子和 4 名男子的集合中选出 k 个人组建一个委员会. 对于下面各情况, 有多少种方法组建这个委员会呢?

(a) 该委员会是由 3 名女子和 2 名男子组成的.

(b) 该委员会的人数不限, 但男女人数必须相等, 且至少各有一人.

(c) 该委员会有 4 人, 而且其中之一必须是 Baggins 先生.

(d) 该委员会有 4 人, 而且至少有两名女子.

(e) 该委员会有 4 人, 男女各有两人, Baggins 夫妇不能同时在这个委员会中.

(a) 对互不相交的集合运用乘法法则, 可以把 3 名女子的所有子集与 2 名男子的所有子集乘起来, 计数 3 名女子和 2 名男子的所有委员会, 于是有 $C(7,3) \times C(4,2) = 35 \times 6 = 210$ 种.

(b) 为了计数这个委员会中女子和男子人数相同的所有可能的子集, 必须明确知道这些子集的大小. 即, 必须把这个问题分解成 4 个互不相交的子集: 1 名女子和 1 名男子, 2 名女子和 2 名男子, 3 名女子和 3 名男子, 4 名女子和 4 名男子 (只有 4 名男子). 所以委员会的总数是这 4 种子情况的可能结果之和. $[C(7,1) \times C(4,1)] + [C(7,2) \times C(4,2)] + [C(7,3) \times C(4,3)] + [C(7,4) \times C(4,4)] = 7\times4+21\times6+35\times4+35\times1 = 329$ 种.

(c) 如果 Baggins 先生必须在委员会中, 那么这个问题就可以简化成从剩余的 10 个人中选出其他 3 个人组建 4 人委员会. 到目前为止, 我们都是从男子集合或者女子集合进行选取. 现在, 其余 3 个人必须从所有剩余的人中选出, 所以答案是 $C(10, 3)$=120 种. (很容易形成这样的习惯思维, 当前问题中使用的集合与前面问题中使用的集合相同, 这里就是男子集合与女子集合, 问题求解者必须时刻提防这样的陷阱.)

(d) 一种方法是首先选出 2 名女子, 有 $C(7, 2)$=21 种方法, 然后从剩余的 9 人 (5 女 4 男) 集合中任意选出 2 名. 然而, 用这种方式来计数委员会数目将导致某些结果不只计数一次, 因为这些委员会成员中的任意一名女子可能是作为前 2 名女子中的一人被选出的, 也可能是作为其余 2 个人中的一人被选出的. 例如, 如果 W_i 表示第 i 个女子, M_i 表示第 i 个男子, 那么 (W_1, W_3) 和 2 名剩余人 (W_2, M_3) 的

组合与 (W_1, W_2) 和 (W_3, M_3) 的组合形成相同的集合.

这一问题的一种正确的解决方案必须使用子情况方法, 如在部分 (b) 中那样. 即, 把这个问题分解成 3 个子情况, 而这些子情况准确地指定委员会中有多少名女子和多少名男子: 2 名女子和 2 名男子, 3 名女子和 1 名男子, 以及 4 名女子. 因此, 答案是 $[C(7,2) \times C(4,2)] + [C(7,3) \times C(4,1)] + C(7,4) = 21 \times 6 + 35 \times 4 + 35 = 301$.

(e) 需要把条件 "Baggins 夫妇不能同时在委员会中" 重述成若干子情况, 在这些子情况中, 我们准确地知道 Baggins 先生和 Baggins 夫人到底谁在委员会中. 注意 Baggins 夫妇都不在委员会的可能性满足所给的条件. 因此要考虑 3 种子情况.

第一种情况是 Baggins 夫人在委员会中, 而 Baggins 先生不在. 这时, 必须从剩余的 6 名女子中再选出 1 名女子, 且从剩余的 3 名男子中再选出 2 名男子 (Baggins 先生被排除在外). 这可以用 $C(6,1) \times C(3,2) = 6 \times 3 = 18$ 种方法完成. 其他两种情况是: Baggins 夫人不在委员会中, 而 Baggins 先生在, 通过类似的推理可以得到 $C(6,2) \times C(3,1) = 15 \times 3 = 45$ 种方法; 两者都不在委员会中, 此时有 $C(6,2) \times C(3,2) = 15 \times 3 = 45$ 种方法. 最后的答案是 18+45+45=108 种方法.

通过取补的方法可以得到这一问题的另一种更简单的解决方案. 考虑所有 $C(7,2) \times C(4,2)$ 种 2 名女子和 2 名男子的委员会, 然后, 减去同时包含 Baggins 夫妇的不符合条件的委员会. 不符合条件的委员会是选出另 1 名女子和另 1 名男子并与 Baggins 夫妇组合而形成的, 所以可以用 $C(6,1) \times C(3,1)$ 种方法完成. 这个方法导致给出有相同答案的更简单公式: $21 \times 6 - 6 \times 3 = 108$ 种方法. □

如例 5 的部分 (d) 所示, 经常会有把相同的结果计数两次的错误. 下面的原理可以帮助读者避开这一问题.

集合合成原理

假设我们使用乘法法则枚举不同对象的集合: 把形成这个集合的某个*第一部分*的方法数与形成 (对于给定第一部分的)*第二部分*的方法数相乘. 第一部分的成员和第二部分的成员必须不相交. 表示这个条件的另一种方式是: 给定如此构造出来的集合 S, 必须能够唯一地告知 S 中的哪些元素在 S 的第一部分, 哪些元素在第二部分.

在例 5(a) 中, 3 名女子的部分 W 和 2 名男子的部分 M 是不相交的, 所以根据集合合成原则, 有 3 名女子和 2 名男子的委员会数量是 W 的大小乘以 M 的大小. 例 5(d) 的问题是 "至少有 2 名女子的委员会" 数量, 先从 7 名女子中选出 2 名女子, 然后再从剩余的人中任意选出两个人 (女子或男子) 的方法违背了集合合成原则, 因为两个部分的成员不是不相交的, 女子可能在第一部分, 也可能在第二部分. 例如, 在委员会 $\{W_1, W_2, W_3, M_3\}$中, 我们无法告知哪两名女子是在第一部分选出的. 选出两名女子, 然后再从剩余的人中任意选出两个人的方案用 $C(3, 2)$ 种不同的方法产生委员会 $\{W_1, W_2, W_3, M_3\}$, 即, 下面两个集合的所有合成: (i)W_1,

W_2, W_3 中的两名女子, (ii)M_3 加上 W_1, W_2, W_3 中没有被 (i) 选中的人.

在例 2 中, 我们讨论了 SYSTEMS 的字母排列, 我们处理了这样的限制: 在排列中某些字母必须被捆在一起. 在例 2 中, 这些字母是 3 个连续的 S. 在这个例子的后续解说中, 我们考虑要求某个特殊字母出现在这个排列中另一个字母之前的限制情况. 我们还会说明如何把这两个限制结合在一起.

例 2(续) 重复排列

单词 SYSTEMS 中的 7 个字母的排列中, E 出现在 M 之前的排列有多少个? E 在 M 之前出现且 3 个 S 挨在一起的排列有多少个?

E 在 M 之前 (不一定与 M 相邻) 出现这一限制的关键是要关注 E 和 M 要进入的一对位置. 因此, 我们先从这个排列的 7 个位置中选出 E 和 M 要进入的一对位置, 有 $C(7, 2)$=21 种方法, 然后, 把 E 放在这一对位置的第一个位置, 而把 M 放在这一对位置的第二个位置. 现在, 为 Y 和 T 选择位置来填充这个排列中的其他 5 个位置, 有 $P(5, 2)=5\times4=20$ 种方法, 再把 3 个 S 放在这个排列中的剩余位置. 因此, 答案是 $21\times20=420$ 种方法.

同时处理这两种限制听起来有些令人恐惧, 事实上, 如果能按正确的顺序处理这些限制, 它们并没有想象的那样困难. 如果首先为 E 和 M 选出一对位置, 那么由于连续性的限制, 事情就会变得混乱, 因为 E 和 M 的不同放置将对排列中有足够空间放置 3 个连续的 S 的位置产生不同的影响.

所以我们尝试着首先处理连续的 S. 这只要求我们把这个 3 个 S“粘”在一起形成一个合成字母, 然后考虑排列 5 个字母 Y, T, E, M, (SSS) 的简化问题. 现在, 再考虑另一个限制, 从 5 个新位置中为 E 和 M 选出一对位置, 有 $C(5, 2)$=10 种方法, 而 E 要进入这两个已选定的位置的第一个位置. 然后, 我们用 3! =6 种方法在剩余的 3 个位置上排列 Y, T, (SSS). 所以答案是 $10\times6=60$ 种方法. □

下面这类计数问题常出现在质量控制领域.

例 6　计数次品数

一家制造厂产生烤箱. 在最后的阶段, 一名检查员要给烤箱标上 A(合格) 或 U(不合格). 15 个 A 和 U 的不同序列中有多少种可能的序列是第 3 个 U 出现在这个序列的第 12 个字母上?

这个问题类似于例 5 中的二进制序列问题, 只是这里元素是 A 和 U, 而不是 0 和 1. 如果第 3 个 U 出在序列中的第 12 个字母上, 那么前 11 个字母组成的子序列中一定有 2 个 U(和 9 个 A). 遵循例 5 中的推理, 对于前 11 个字母, 有 $C(11,2) = 55$ 种可能的子序列. 只有一种方法选择第 12 个字母, 因为这个字母是指定的 U. 这个序列的剩余 3 个字母可以是 A 或 U, 所以有 $2^3 = 8$ 种可能. 所以, 总共有 $55\times1\times8 = 440$ 种可能的序列. □

例 7　重复数字的概率

4 位数大学校园电话号码中有一个或多个重复数字的概率是多少?

有 10^4=10 000 种不同的 4 位数电话号码. 把这个重复数字 4 位电话号码的计数问题分解成为 4 种不同的重复情况:

(a) 所有 4 个数字都相同.

(b) 3 个数字相同, 另 1 个数字不同.

(c) 2 个数字相同, 另 2 个数字也相同 (例如, 2 828).

(d) 2 个数字相同, 另 2 个数字互不相同 (例如, 5 105).

(a) 由 10 个数字之一重复 4 次可以形成 10 个号码.

(b) 分解形成情况 (b) 的这些号码的一种方法如下 (有若干其他分解方法). 首先挑出只出现一次的数字, 有 10 种选择, 然后它出现在 4 位号码中的位置, 有 4 种选择, 最后挑选出现在其他 3 个位置上的另一个数字, 有 9 种选择. 因此, 有 $10\times4\times9=360$ 种号码.

(c) 首先选出 2 个数字, 每一个都出现 2 次, 有 $C(10,2)=45$ 种选择, 然后是如何排列这 4 个数字: 比如说为较小的数字选出 2 个位置, 有 $C(4,2)$=6 种方法. 这产生 $45\times6=270$ 个号码.

(d) 首先挑出只出现一次的一对数字, 有 $C(10,2)=45$ 种选择, 然后为其中较小的数字选出一个位置, 再为其中较大的数字选出一个位置, 有 $4\times3=12$ 种选择, 最后从剩余的数字中选出一个数字占据剩下的 2 个位置, 有 8 种选择. 因此产生 $45\times12\times8=4\ 320$ 种号码.

总之, 共有 10+360+270+4 320=4 960 个有重复数字的 4 位电话号码. 因此, 有重复数字的电话号码的概率是 4 960/10 000=0.496≈0.5.

注意, 还有一种更简单的方法, 利用补集来计数有重复数字的电话号码: 计数没有重复数字的电话号码. 没有重复的电话号码数是 10 个数字的 4 置换, 有 $P(10,4)=5\ 040$ 种. 所以剩下的重复数字电话号码的数量是 10 000−5 040=4 960 种. □

对于情况 (c) 和 (d) 有一点要注意, 在这两种情况中, 两个不同的数字都出现一次或者都出现两次. 在情况 (d) 中, 选出 2 个出现一次的数字, 作为无序对, 有 $C(10,2)$ 种方法, 然后排列这 2 个数字 (并选出 1 个数字进入剩余 2 个位置), 而不是采用这样的方法: 先选出第一个数字, 放置它, 然后选出第二个数字, 放置它 (然后选出一个数字进入剩余的 2 个位置), 共有 $10\times4\times9\times3\times8$ 种方法. 在后面的这一 (错误的) 方法中, 我们无法告知诸如 2 529 这样的号码是首先选出 5, 把它放置在第 2 个位置上, 然后再选出 9, 把它放置在第 4 个位置上, 还是先选出 9, 把它放置在第 4 个位置上, 然后选出 5, 把它放置在第 2 个位置上. 这违背了乘法法则的互不相交要求, 情况 (d) 中每一个结果都被计数两次.

例 8　投票人势力

考虑评估加权投票中不同参与者的影响力的方法. 假设在一个 5 人地方理事会中, 有 3 个代表来自小镇, 称他们为 a, b, c, 他们每个人投一票, 其他 2 个代表来

自较大的镇, 称他们为 D, E, 他们每个人投两票. 由于总共有 7 张选票, 同意立法需要 4 张票 (过半数的票). 假设在形成一个联盟对某项立法投票时, 这些人按顺序加入这个联盟 (他们的一个排列). 这个联盟排列中的枢轴 (pivotal) 人是这样的代表, 他的投票使这个联盟中的投票数达到 4. 例如, 在联盟排列 bDcaE 中, 枢轴人是 c. 理事会中的人 p 的投票 "势力" 表示为 p 是枢轴人物的联盟排列占所有联盟排列的比例. 这一势力度量称为 Shapley-Shubik 指数 (Shapley-Shubik index).

在这个理事会中确定人 a 和 D 的 Shapley-Shubik 指数, 即确定在所有联盟排列中 a 和 D 分别是枢轴人物的比例. (根据对称性, 1 张选票的人 b 和 c 与 a 有相同的指数, 类似地, E 与 D 有相同的指数.)

如果 a 在某个联盟排列中是枢轴人, 那么 a 前面的人必须正好有 3 张票, a 后面的人有 3 张票. 因为仅有其他两个投 1 票的人, 所以在 a 前面的 3 票一定是一个 1 票 (b 或 c) 和一个 2 票 (D 或 E). 于是, 可以有 2(1 张选票投票人的选择)×2(2 张选票投票人的选择)×2(1 张选票投票人或者 2 张选票投票人哪个在前)=8 种方法形成这个联盟的前面部分. 其余 1 张选票投票人和 2 张选票投票人在 a 之后, 有 2 种排列他们的方法. a 是枢轴人的联盟排列总共有 8×2=16 种. 一共有 5! =120 种排列, 所以 a 的 Shapley-Shubik 指数是 16/120=4/30.

如果 D 是联盟排列中的枢轴人, 那么在 D 前面的人或者有 2 张票或者有 3 张票. 假设在 D 之前有 2 张票, 而在 D 之后有 3 张票. 于是, 这个排列或者是以 3 个 1 张选票投票人中的 2 个人开始, 排列他们有 3×2=6 种方法, 然后是 D, 之后是另外一个 1 张选票投票人和 E, 有 2 种方法, 或者是这个排列以 E 开始, 然后是 D, 随后是 3 个 1 张选票投票人的全排列, 有 3! 种方法. 总共有 (3×2×2)+3!=18 种排列, 在这些排列中, D 之前有 2 张票, D 之后有 3 张票. 在这些排列中把 D 之前的人与 D 之后的人交换, 我们得到 D 之前有 3 张票, D 之后有 2 张票的排列. 所以后一种类的排列有 18 种. 求和得, D 是枢轴人的排列总共有 18+18=36 种, 所以 D 的 Shapley-Shubik 指数是 36/120=9/30.

观察可知, 有 2 张票的投票人的指数是 1 张票投票人的指数的 $2\frac{1}{4}$ 倍. □

我们以下面的内容结束本节, 对于充分大的 n, $n!$ 可以由数 $s_n = \sqrt{2\pi n}(n/\mathrm{e})^n$ 近似表示, 其中 e 是欧拉常数 ($\mathrm{e} = 2.718\ldots$). 这个近似值是由 Stirling 发现的, 其推导在很多高级微积分教科书中都可以找到 (参见 Buck[1]). 误差 $|n! - s_n|$ 随着 n 的增大而增大, 但是相对误差 $|n! - s_n|/s_n$ 总小于 $1/11n$. 下表给出若干 $n!$, s_n, $|n! - s_n|/s_n$ 的样例值.

n	$n!$	s_n	$\lvert n! - s_n\rvert/s_n$
1	1	0.922	0.085
2	2	1.919	0.042
5	120	118.02	0.017

(续)

n	$n!$	s_n	$\|n! - s_n\|/s_n$
10	3 628 800	3 598 600	0.008
20	2.433×10^{18}	2.423×10^{18}	0.004
100	9.333×10^{157}	9.328×10^{157}	0.000 5

练 习 5.2

练习概况 如 5.1 节一样, 大多数练习都需要具体分析, 对于每一个问题分析都不相同. 记住, 要认真阅读练习以防曲解. 要格外注意问题是关于排列还是子集的. 注意问题假设人是不同的对象 (没有相同的人).

1. 有多少种方法排列一副 52 张纸牌?

2. 字母表中 26 个字母可以形成多少没有重复的 8 字母 "单词"(序列)?

3. 在 15 个孩子间分配 7 本不同的书, 如果孩子最多只得到 1 本书, 那么有多少种不同的分法?

评价下面的答案: **(a)** $C(15, 7)$; **(b)** 15^7; **(c)** $C(15, 7)7!$.

4. 沿着一边有 8 个座位的长桌子的一侧给 4 名男孩子和 4 名女孩子分配座位, 有多少种不同的分法? 如果男孩子和女孩子交替落座, 有多少种方法?

5. 单词 VISITING 中的 8 个字母有多少种不同的排列?

6. 有多少种方法从字母表的 26 个字母中挑出一个 6 个不同字母的子集?

7. 有多少种方法从 12 名选手中挑选出一个 5 人篮球队? 如果最弱的选手和最强的选手必须在这个队里, 那么有多少种方法?

8. 一个缸里有 8 个白球和 5 个红球. 假设 13 个球是不同的, 有多少种方法选出一个有 5 个球的子集? 选出 3 个白球和 2 个红球的概率是多少?

9. 如果投掷一枚均匀的硬币 9 次, 有 7 次或更多次正面的概率是多少?

10. 在字母 a, b, c, e, f, g 的排列中, 以元音开头和结尾的排列的概率是多少?

11. 给定 5 双 (共 10 只) 不同的手套, 给 5 个姐妹中的每个人分配两只手套, 有多少种方法?

(a) 如果某个人得到的两只手套都是左手或者都是右手.

(b) 如果每个姐妹都得到一只左手手套和一只右手手套.

12. 有多少种方法把 14 人分成:

(a) 大小分别为 3, 5, 6 的三组? **(b)** 大小为 7 的两组 (无序)?

13. 正好有 1 个 A 和 2 个 B 的 (由字母表中的 26 个字母形成的且允许重复的)5 字母序列有多少?

14. **(a)** 在一次有 10 个问题的测试中, 正好正确地回答 8 个问题有多少种方法?

(b) 增加一个要求, 第一题或者第二题的回答正确, 但是二者不同时回答正确时, 重复部分 (a).

(c) 在前 5 问中的 3 个问题回答正确的情况下, 重复部分 (a).

15. 有多少正好有 9 个 0 的 n 位三进制 (0, 1, 2) 序列?

评价下面的答案: **(a)** 3^{n-9}; **(b)** 2^{n-9}; **(c)** $C(n, 8)3^{n-9}$.

16. 5 张牌组成的一手牌有下面各情况的概率是多少?

(a) 4 张 A. (b) 四同号. (c) 两对子 (不是四同号, 也不是满堂彩).
(d) 满堂彩 (3 张同号牌外加一个对子). (e) 顺号 (5 张连续值的一组牌).
(f) 没有对 (可能是顺号或满堂彩).

17. 由 4 名男子和 6 名女子组建满足下面条件的委员会, 有多少种方法:
(a) 至少两名男子且女子人数至少是男子人数的两倍?
(b) 委员会的人数是 3 到 5 人, 而且 Wonder 小姐不在委员会中.
(c) 5 名成员, 而且 O'Hara 三姐妹不能都在委员会中.
(d) 4 名成员, 其中至少有 2 名女子, Baggins 夫妇不能同时在委员会中.

18. 假设一个 50 人的班级有 20 名男生和 30 名女生. 有 3 门不同的全国竞赛: 数学、化学和英语, 指导员要为每项竞赛挑出 1 名 (不同的) 学生参加.
(a) 正好选中 1 名女生的概率是多少?
(b) 至少选中 2 名男生的概率是多少?

19. 假设校园电话号码是由任意 4 位数字组成的 (允许重复).
(a) 数字 6 在电话号码中至少出现两次的概率是多少?
(b) 一个号码正好包含 2 个不同的数字 (例如: 2 444) 的概率是多少?
(c) 一个号码是由 4 个按递增顺序排列的数字组成的概率是多少? (例如: 2 578)

20. 有 8 名候选人申请捕狗手的工作, 3 名评判官给候选人评分. 候选人被选中当且仅当他们在所有 3 个评判官给出的定级中均名列前 3.
(a) 3 名评判官有多少种方法给出他们的 3 个定级?
(b) 在 3 个定级的随机集合中, 候选人之一的 Dickens 先生被选中的概率是多少?

21. 有 6 本不同的法语书, 8 本不同的俄语书, 5 本不同的西班牙语书. 有多少方法把这些书在书架上排成一行, 使得语种相同的书在一起?

22. 如果 13 个玩家中的每一个玩家都从一副 52 张的纸牌中抽出 4 张, 每一个玩家都得到 4 种花色的牌的概率是多少?

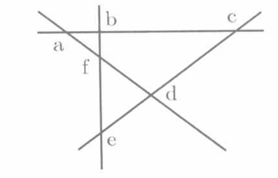
23. 使用 5 个不同的元音和 5 个 (从 21 个辅音中选出的) 不同辅音, 可以形成多少不同的 10 字母序列? 其中没有连续辅音对的单词的概率是多少?
评价下面的答案: **(a)** $3(n-1)!$; **(b)** $(3!\,(n-3)\,!)^3$; **(c)** $3((n-1)\,!)^3$.

24. 随机的 9 位社会保险号码至少有一位重复数字的概率是多少?

25. 字母 a, b, c, d, e, f 的排列中有下面各性质的排列的概率是多少:
(a) a 和 b 挨在一起? (b) a 在 b 之前出现?

26. 在一次舞会上, 给 10 名女子与 20 名男子中的 10 名男子配对有多少种方法?

27. 根据下面的条件, 可以形成多少个三角形 (假设没有三线共点的情况):
(a) n 条非平行线形成多少三角形? 例如, 4 条直线可以形成 4 个三角形: acd, abf, efd, ebc.
(b) 有 n 条直线, 其中有 m 条直线平行, 其他两两不平行, 可以形成多少个三角形?



28. 有 3 名女子和 5 名男子要组成 2 个 4 人队伍. 有多少种方法完成这一分组, 使得每个队伍中至少有 1 名女子?

29. 一名男子有 n 个朋友, 而且一年中 (365 天) 每一个晚上都邀请其朋友的不同 4 人组到他

家. 这个 n 必须多大?

30. JUPITER 的排列中有多少个其元音顺序是字母表顺序的排列?
评价答案: **(a)** 3!4!; **(b)** 7!/3!; **(c)** $C(7, 3)4!$.

31. 在由下面人员组成的理事会中, 确定 1 张投票权的人和 2 张投票权的人的 Shapley-Shubik 指数:
(a) 3 名 1 张投票权的人和 1 名 2 张投票权的人.
(b) 2 名 1 张投票权的人和 2 名 2 张投票权的人 (4 张票是过半数).
(c) 4 名 1 张投票权的人和 2 名 2 张投票权的人 (6 张票是过半数).
(d) 4 名 1 张投票权的人和 4 名 2 张投票权的人 (7 张票是过半数).

32. 其中有 3 个对子的 7 张一手牌 (每个对子的点数不同, 加上第 7 张不同点数的牌) 有多少?

33. 满足下面条件的 26 个字母的排列有多少种?
(a) a 在 b 前面出现. **(b)** a 在 b 的前面出现, 且 c 在 d 的前面出现.
(c) 5 个元音按字母表顺序出现.

34. 在一次测试中, 学生必须回答 10 个问题中的 5 个问题, 且包含前 5 道题中的至少两道题. 有多少个学生可以回答的 5 题子集?

35. 使用至少 3 个元音 (A, E, I, O, U), 可以形成多少个 6 字母序列? 要求没有重复.

36. 1, 1, 1, 1, 2, 3, 3 的排列中有多少个 2 不在 3 旁边的排列?

37. INSTRUCTOR 字母的排列中, 两个连续的元音之间正好有两个辅音的排列有多少个?

38. 重新排列 INQUISITIVE, 使得 U 不挨在 Q 的后面, 这样可以形成多少单词?

39. 假设在一次抽奖中, 随机选出一年中含有 60 个不同日子的一个子集. 在这个子集中, 每一个月份有 5 天的概率是多少? (为了简单起见, 假设一年有 12 个月, 每个月有 30 天.)

40. 假设随机选出一年中的 8 天. 每一天来自不同的月份的概率是多少? (为了简单起见, 假设一年有 12 个月, 每个月有 30 天.)

41. 从 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 这 10 个数字的置换中, 随机选出满足下面条件的置换的概率是多少:
(a) 第一个位置是奇数, 且最后位置是 1, 2, 3, 4, 5 之一?
(b) 5 不在第一个位置, 且 9 不在最后位置?

42. **(a)** 一手 5 张牌中每一种花色至少一张的概率是多少?
(b) 对于一手 6 张牌, 重复上面的问题.

43. 一手 5 张牌满足下面条件的概率是多少:
(a) 下面 4 个值每种至少有一张: A, K, Q, J.
(b) 红心和黑桃的数量相同.

44. 有 k 个 1 的 n 位三进制数序列有多少个?

45. INSTRUCTOR 的所有排列中符合下列条件的排列的比例是多少:
(a) 3 个连续元音? **(b)** 2 个连续元音?

46. 如果整数 $1, 2, \cdots, m$ 的所有 3 元子集的 $\frac{1}{4}$ 包含 5, 确定 m.

47. 有多少种方法从 30 个人中选出 4 对 2 人组?

48. GRACEFUL 的所有排列中, 符合下面条件的排列的比例是多少:
(a) F 和 G 挨在一起? **(b)** 没有挨在一起的元音对?

49. SYSTEMATIC 的排列中, 每一个 S 的后面是一个元音 (元音包括 Y) 的排列有多少个?

50. MATHEMATICS 的排列中, 每一个辅音都与一个元音相邻的排列有多少个?

51. **(a)** 从 1 到 20 之间选出的 4 个不同的数, k 是最小整数的概率是多少 ($1\leqslant k \leqslant 17$)?
(b) k 是第二小整数的概率是多少?

52. 从 1 到 20 之间选出 4 个不同的数, 最大整数和最小整数的差是 k 的概率是多少 ($3\leqslant k \leqslant 19$)?

53. 把 8 个相同的黑子和 8 个相同的白子放置在一个 8×8 的棋盘上有多少种方法?

54. 由字母表中的 26 个字母形成的 (不重复)8 字母排列中, 最多包含 5 个元音中的 3 个元音且元音以字母表顺序出现的排列有多少个? (提示: 分解成子情况.)

55. 在随机的一组 25 个人当中, 有 2 个或更多人有相同生日的概率是多少? (这是著名的**生日悖论问题**.)

56. 有多少种方法安排 n 个 (不同) 人, 使得: (i)Smith 夫妇挨在一起; (ii)Tucker 夫人距离 Smith 夫妇有 k 个位置 (即, Tucker 夫人与 Smith 夫妇之间有 k-1 个人)?

57. 一支篮球队有 5 名选手, 其中 3 个人在 "前锋" 的位置 (包含 "中锋"), 2 个人在 "后卫" 的位置. 如果有 6 个前锋, 4 个后卫, 2 个人既可以做前锋也可以做后卫, 有多少种方法选出这样一支队伍?

58. 一个家庭要将 2 个男孩和 3 个女孩送到私立学校. 有 5 所男子学校、8 所女子学校, 以及 3 所男女合校. 如果每个孩子去不同的学校, 这个家庭为他们的孩子可以选择多少组不同的 5 所学校?

59. 给定一组 $2n$ 个对象, 有 n 个对象相同, 另 n 个对象互不相同, 可以形成多少组不同的 n 个对象?

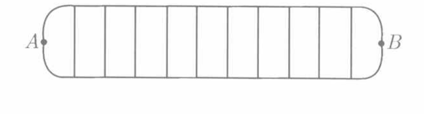
60. 一批 50 台不同的自动打字机正好包含 10 台次品. 找到满足下面条件的事件的概率:
(a) 在随机取出的 5 台机器中至少有 1 台是次品.
(b) 在随机取出的 10 台机器中至少有 2 台是次品.
(c) 在机器的随机检查序列中, 第一台次品正好是参加检查的第 k 台机器.
(d) 最后一台次品正好是参检的第 k 台机器.

61. 在一个湖里捕到 10 条鱼, 做上标记, 然后再放生回湖里. 两天后, 又捕到 20 条鱼, 其中有 2 条已经做了标记.
(a) 如果这个湖里有 k 条鱼, 那么寻找 20 条鱼中 2 条鱼做了标记的概率. (假设鱼是随机捕到的.)
(b) 使这个概率最大的 k 值是多少?

62. Grinch 教授的电话号码是 6328363. Mickey 记住了这组数字, 但是顺序不对, 但是他知道第一个 6 在第一个 3 之前. 满足这一限制的这组数字的排列有多少个?

63. PREPOSTEROUS 中的字母排列中, 第一个出现的元音是 E 的排列有多少个?

64. 对于下面给定的城市 A 和城市 B 之间的道路图,



(a) 从城市 A 到城市 B 不重复任何道路的线路有多少条? (道路是两个交叉点之间的线段.)
(b) 对于要从城市 A 到城市 B 的两个人来说, 有多少方法使得他们二人从没有在同一方向上经过相同道路?

65. 随机取出一手 5 张牌, 满足下面条件的概率是多少:

(a) 正好有一对 (没有 3 张同点数牌或两对子)?
评价下面的答案: **(a)** $13\times C(4,2)\times C(48,3)/C(52,5)$, **(b)** $13\times C(4,2)\times 48\times 44\times 40/C(52,5)$, **(c)** $(52\times 3\times 48\times 44\times 40)/(52\times 51\times 50\times 49\times 48)$.

(b) 一对或更好 (同点数的 3 张牌, 两对子, 四同号, 满堂彩).

(c) 牌按值的递减顺序入手.

(d) 至少有一张黑桃和一张红心, 没有方块或梅花, 而且黑桃的值都比红心的值大.

66. 有多少种方法从 100 个人 (每个人的身高都不同) 选出一组 n 个人, 然后再挑出第二组 m 个人, 使得第一组人都比第二组人的身高高?

67. 1 到 90 之间有多少组 3 个不同的整数, 这 3 个整数的和是:
(a) 一个偶数?　**(b)** 可以被 3 整除?　**(c)** 可以被 4 整除?

68. 从 MISSISSIPPI 中选出两个不同字母的概率是多少? 选出 3 个完全不同的字母的概率是多少?

69. 如果投掷一枚硬币 8 次, 得到下面事件的概率是多少:
(a) 正好有 4 个正面?　**(b)** 至少有 4 个正面?

70. 一名女子在连续的 3 天内邀请她 5 个朋友中的一组 3 个朋友, 有多少种方法? 如果她有 n 个朋友, 又有多少种方法?

71. 连接凸 8 边形的不同的 3 顶点组, 可以形成多少个不同的三角形? 如果不允许有一对相邻的顶点, 又有多少个三角形?

72. 5 个 0 和 10 个 1 所形成的排列中, 没有连续 0 对的排列有多少个?

73. **(a)** 凸 n 边形的弦可以形成多少个交点? (假设这些直线中每 3 条线都不相交于一点.)
(b) 这些交点把 (a) 中的直线分成多少条线段?
(c) 使用欧拉公式 $r=e-v+2$ 以及 (a) 和 (b) 的结果, 确定由 n 边形的弦形成的区域数.

74. 给定 14 个正整数 (其中 12 个是偶数), 及 16 个负整数 (其中 11 个是偶数). 有多少种方法从这 30 个数中挑出 12 个数, 使得 12 个数中的 6 个是正数, 且 12 个数中的 6 个是偶数?

75. 下面各情况可以形成多少个三角形 (假设 3 条直线都不相交于一点):
(a) 凸 10 边形的 3 条弦形成的完全在这个 10 边形内的三角形 (10 边形的顶点不能是其中任意三角形的顶点).
(b) 一个凸 n 边形的 3 条弦和外面的 10 条边可以形成多少个三角形?

76. 整数 $1, 2, \cdots n$ 所形成的排列中, 每一个整数 (第一个整数除外) 都与这个排列中它左边的某个整数的差是 1 的排列有多少个?

77. 一个人有 7 个朋友. 他有多少种方法在连续 7 天内, 邀请这 7 个朋友中的不同 3 人组参加晚宴, 使得每一对朋友正好在某一次晚宴上相遇?

5.3　重复排列和选取

本节讨论带有重复的排列和选取, 即诸如 b, a, n, a, n, a 这样有重复对象的集合的排列, 以及一个对象可以被选择多次的选取, 例如从 3 种热狗中取出 6 个的选择. 我们利用刚刚提到的两个例子导出这些计数问题的公式.

例 1 banana 的排列

6 个字母 b, a, n, a, n, a 的排列有多少种?

考虑可能的排列:

$$\underline{n}\ \underline{a}\ \underline{b}\ \underline{n}\ \underline{a}\ \underline{a}$$

我们扩展有 6 个 1 的 8 位二进制序列 (5.2 节的例 3) 的计算问题中所使用的推理来解决这一问题. 其关键是注意 a 进入的位置的子集和 n 进入的位置的子集. 例如, 上面的排列是由 a 在位置 2, 5, 6 上, n 在位置 1, 4 上, b 在位置 3 上所刻画的. 我们如下计数这些排列: 首先, 选出排列中 a 将进入的三个位置的子集, 有 $C(6,3) = 20$ 种方法; 然后, 从剩余的位置中选出 n 将进入的两个位置的子集, 有 $C(3,2) = 3$ 种, 最后剩余的位置由 b 得到, 有 1 种方法. 因此有 $C(6,3) \times C(3,2) \times C(1,1) = 20 \times 3 \times 1 = 60$ 种排列. □

定理 1 如果有 n 个对象, 类型 1 的对象有 r_1 个, 类型 2 的对象有 r_2 个, $\cdots$, 类型 m 的对象有 r_m 个, 其中 $r_1 + r_2 + \cdots + r_m = n$, 那么这 n 个对象的排列数记为 $P(n; r_1, r_2, \cdots, r_m)$, 等于

$$\begin{aligned} P(n;r_1,r_2,\cdots,r_m) &= \binom{n}{r_1}\binom{n-r_1}{r_2}\binom{n-r_1-r_2}{r_3}\cdots\binom{n-r_1-r_2\cdots-r_{m-1}}{r_m} \\ &= \frac{n!}{r_1!r_2!\cdots r_m!} \end{aligned} \tag{$*$}$$

证明 1 首先为第一种类型选出 r_1 个位置, 然后为第二种类型选出 r_2 种位置, 以此类推. 省略部分的精确数学证明需要归纳法 (参见练习 35). $(*)$ 的第二行是消去二项式系数中的因子后得到的简化结果, 例如,

$$P(6;3,2,1) = \binom{6}{3}\binom{3}{2}\binom{1}{1} = \frac{6!}{3!3!} \times \frac{3!}{2!1!} \times \frac{1!}{1!} = \frac{6!}{3!2!1!} \qquad \diamondsuit$$

证明 2 这一证明类似于利用等式 $P(n,r) = C(n,r)P(r,r)$ 导出 $C(n,r)$ 时的推导. 假设对于每一种类型, 把类型 i 的 r_i 个对象赋下标, 标记为 $1, 2, \cdots, r_i$, 使得每个对象不同. n 个对象有 $n!$ 个排列. 我们通过枚举出这些对象的所有 $P(n; r_1, r_2, \cdots, r_m)$ 个 (没有下标的) 模式, 然后再对于每一个模式, 用所有可能的方法放置下标, 这样来枚举出 n 个不同对象的所有 $n!$ 个排列. 例如, 模式 baanna 可以有 3! 种给 a 赋下标的方法:

$$\begin{array}{lll} \mathrm{ba_1a_2nna_3} & \mathrm{ba_2a_1nna_3} & \mathrm{ba_3a_1nna_2} \\ \mathrm{ba_3a_2nna_1} & \mathrm{ba_1a_3nna_2} & \mathrm{ba_2a_3nna_1} \end{array}$$

对于给 a 赋下标的这 3! 种方法中的每一种方法, 有 2! 种给 n 赋下标的方法. 因此, 一般地, 在一个模式中, 有 $r_1!$ 种给类型 1 的 r_1 个对象赋下标, $r_2!$ 种给类型 2 的 r_2 个对象赋下标, $r_m!$ 种给类型 m 的 r_m 个对象赋下标的方法. 于是

$$n! = p(n; r_1, r_2, \cdots, r_m) r_1! r_2! \cdots r_m!$$

或者

$$P(n; r_1, r_2, \cdots, r_m) = \frac{n!}{r_1! r_2! \cdots r_m!} \qquad \diamond$$

例 2　排序热狗

有多少种不同的方法从 3 个品种的热狗中选出 6 个热狗?

为了解决这样的带有重复的选取问题, 我们如下把这一问题重述成一个排列问题. 假设这 3 个品种分别是普通热狗、辣椒热狗和高级热狗. 按着下面的方式把一种选取按有序形式书写下来 (当有人按这种顺序放置时):

普通热狗	辣椒热狗	高级热狗
x	xxxx	x

每一个 x 代表一个热狗. 上图所展示的要求是 1 个普通热狗、4 个辣椒热狗和 1 个高级热狗. 因为热狗厨师知道这种形式的热狗序列是普通、辣椒和高级热狗, 这种要求可以简单地写成 x ｜ xxxx ｜ x, 去掉表头.

r 个热狗的选取是由 r 个 x 和两个 | 的某个序列组成的. 反过来, r 个 x 和两个 | 的任意序列代表唯一一种选取：第一个 | 之前的若干个 x 计数的是普通热狗数, 两个 | 之间的 x 计数的是辣椒热狗数, 最后的若干个 x 计数的是高级热狗数. 所以热狗顺序与这样的序列之间存在一一对应. 6 个 x 和两个 | 的序列的计数是重复排列问题, 其答案是 $P(8;6,2) = \binom{8}{6}\binom{2}{2} = \frac{8!}{6!2!}$. 如 5.2 节例 3 中的讨论那样, 若干个 x 和若干个 | 的序列的计数就是简单地挑出 x (或者是 |) 将要进入的位置的子集数, 所以答案仍然是 $\binom{8}{6}$ 种方法. □

定理 2　*从 n 个类型的对象中选出 r 个对象的重复选取数量是 $C(r+n-1, r)$.*

证明　我们像例 2 中那样给一种选取赋予一种 "顺序形式", 以 x 代表每一个选出的对象. 正如前面描述的那样, 第一个 | 之前的若干个 x 计数的是第一种类型的对象的数量, 第一个和第二个 | 之间的若干个 x 表示第二种类型对象的数量, 第 $(n-1)$ 个 | 之后的 x 表示第 n 种类型对象的数量 (隔开 n 个类型需要 $n-1$ 条竖线). 有 r 个 x 和 $n-1$ 个 | 的序列数量是 $C(r+(n-1), r)$. ◇

例 3　班级分组

一共有 9 个学生, 其中 3 个来自 A 班, 3 个来自 B 班, 3 个来自 C 班, 这 9 个人已经得到有 9 个座位的区域来进行学校组织的校庆活动. 如果从一排 9 个座位中随机为每个班级选出 3 个座位, 那么 3 个 A 班学生、3 个 B 班学生、3 个 C 班学生都得到 3 个连续座位的概率是多少?

在概率问题中, 我们要求理想结果数除以总结果数. 第一个问题是, 在这个问题中所有结果集合是什么? 一共有 $P(9;\ 3,\ 3,\ 3)=9!/3!3!3!=1\ 680$ 种方法在一排 9 个座位上安排 3 个 A、3 个 B 和 3 个 C.

如果每个班的 3 个学生都要坐在一起, 那么我们想计数的结果是 3 个块 AAA, BBB, CCC 的排列. 因此, 取代 9 个字母的排列, 我们实际上是要进行 3 个合成

字母的排列. 共有 3!=6 种方法安排 3 个合成字母. 所以每个班坐在一起的概率是 6/1 680. □

例 4 排列基因

有机体的基因代码存储于 DNA 分子, 而 DNA 分子是由 A(腺嘌呤), C(胞嘧啶), G(鸟嘌呤), T(胸腺嘧啶) 四种核苷酸组成的长链. DNA 的短链可通过各种现代生物学方法 “排序”, 即它由字母序列确定. 尽管单个基因的 DNA 序列就有成百上千万个字母, 但是仍然存在特殊的酶, 它能把一个长链分成短片段 (它们能够被排列), 其方法就是把这个长链在某个特定字母的每一次出现的后面断开.

假设 C- 酶 (它在 C 的每一次出现的后面断开) 把一个 20 个字母的链断开成 8 段, 其每一段确定为: AC, AC, AAATC, C, C, C, TATA, TGGC. 注意, 除了这个链的最后一段之外, 每一段必须以 C 结尾. 这样的片段集合可以生成多少不同的链?

因为片段 TATA 不以 C 结尾, 所以它一定是这个链的结尾. 其他 7 个片段可以以任意顺序出现. 这些片段是由 3 个 C、2 个 AC、1 个 AAATC 和 1 个 TGGC 组成的. 类似于例 3 中的推理, 我们把每一个片段当作一个字母处理. 于是, 我们必须排列 7 个字母, 第一个类型有 3 个, 第二个类型有 2 个, 其他两个类型各有 1 个. 因此, 形成 20 字母的链共有 $P(7;3,2,1,1)=420$ 种这些片段的可能排列. □

如果使用 A 酶把同样的链断开并观察这些片段的所有可能的排列, 然后对 G 酶和 T 酶也做同样的事情, 那么通常只有一个链出现于所有 4 种排列中. 这就是原来的长链. 通常使用这一方法的更加复杂的版本来确定整个基因的 DNA 序列.

例 5 重复度可变的序列

由 4 个 a、4 个 b、4 个 c 和 4 个 d 可以形成多少个每个字母至少出现 2 次的 10 字母序列?

为了运用定理 1, 需要精确地知道 a, b, c, d 将在序列中出现的次数. 因此, 我们必须把这一问题分解成每一个都包含给定数量的 a, b, c, d 的序列的一组子问题. 每个字母出现两次或更多次, 且所有字母出现的次数之和等于 10 的字母出现频率有两种类型. 第一种类型是一个字母出现 4 次, 而其他每个字母出现 2 次. 第二种类型是 2 个字母出现 3 次, 而其他 2 个字母各出现 2 次.

对于第一种类型, 选择哪一个字母出现 4 次共有 4 种情况, 因而有 $P(10;4,2,2,2)$ =18 900 种方法排列一个字母出现 4 次而其他 3 个字母各出现 2 次的序列. 对于第二种类型, 共有 $C(4,2)=6$ 种方法选择哪 2 个字母出现 3 次, 因而有 $P(10;3,3,2,2)$ =25 200 种方法排列 2 个字母出现 3 次而其他 2 个字母出现 2 次的序列. 最后的答案是 4×18 900+6×25 200=226 800 种方法. □

例 6 选取面包圈

从 5 个不同品种的面包圈中挑出 12 个面包圈来装满一个盒子, 且要求每个品种必须至少选出 1 个时, 有多少种方法呢?

我们通过展示人们挑出这样的面包圈的典型过程来解决这个问题. 他们可能首

先从每个品种中选 1 个面包圈, 然后以他们喜欢的方式挑出剩余的 7 个面包圈. 在从每一种类型选出 1 个面包圈时, 没有选择 (只有 1 种方法). 从 5 种类型中选出其余的 7 个面包圈时才存在选择问题. 想象把每一种类型的一个面包圈放入盒子里, 然后用蜡纸把它们盖上, 再选出其余 7 个面包圈. 结果的变化只涉及放到蜡纸上面的面包圈. 所以根据定理 2, 答案是 $C(7+5-1,7)=330$ 种选择. (注意这里不能使用集合合成原理, 因为对象不是完全不相同的.) □

注意, 在重复选取中, 我们只关心计数对于每种类型有多少种选取方法. 运用这一目标, 处理这些类型的下界比较容易 (如在例 6 中那样), 因为这样的限制把这一问题转述为只计数比每一种类型的下界多多少. 然而, 对于重复排列, 这样的计数问题就要复杂的多. 对于重复选取问题中的每一个结果, 即, 在考虑每一种类型的特定数量时, 我们需要处理计数有多少种方法排列这个选择中的各项这样的子问题.

例 7　有下界和上界的选取

有一堆球, 其中有红球、蓝色和紫球, 有多少种方法从这堆球中挑出正好 10 个球且至少有 5 个红球呢? 如果至多有 5 个红球, 又有多少种方法呢?

第一个问题类似于例 6. 我们把 5 个红球放入一个盒子里, 用蜡纸把它们盖上, 然后再随机地选出 5 个球 (有可能再包含红球). 蜡纸上面的 5 个球可以用 $C(5+3-1,5)=21$ 种方法取出.

当某个对象在选取中的重复数存在上界时, 就没有直接计数选取数的方法了. 为了处理至多有 5 个红球的限制, 我们计数补集. 无限制地从 3 种颜色的球中选出 10 个球的方法数是 $C(10+3-1,10)=66$, 选出至少有 6 个红球的集合有 $C(4+3-1,4)=15$ 种方法 (把 6 个球放入一个盒子里, 用蜡纸盖上, 然后再随机地挑出 4 个球). 所以有 66−15=51 种方法选出 10 个球, 其中红球数不超过 5. □

例 8　限制位置排列

回到例 1 关于 banana 的字母的排列, 但是现在要附加 5.2 节中所提出的某些限制. 满足下面限制的字母 b, a, n, a, n, a 的排列有多少个:

(a) b 的后面紧挨着一个 a: 使用 5.2 节例 2 所介绍的计数有连续字母的排列方法; 即把 b 与其中一个 a 粘合到一起形成复合字母 ba. 现在要计数下面 5 个 “字母” 的所有排列: ba, a, a, n, n. 根据定理 1, 总共有 5!/1!2!2!=30 种排列.

(b) 模式 bnn 从不出现: 首先我们无限制地计数字母 b, a, n, a, n, a 的所有排列, 然后减去有模式 bnn 的排列, 这样就可以解决这个问题. 处理有这个模式的排列的方法与部分 (a) 中处理 ba 的方法一样. 我们把 bnn 当作一个合成字母, 因此计数 4 个字母 bnn, a, a, a 的排列, 有 4!/1!3!=4 种方法. 从所有排列的 6!/1!3!2!=60 种方法中减去不符合限制条件的 4 种排列就得到答案: 60−4=56 种.

(c) b 在所有 a 之前出现 (不一定与 a 相邻): 换句话说, b 和 3 个 a 的相对顺序是 b-a-a-a. 我们使用 5.2 节例 2 的续中所介绍的处理相对顺序的方法. 在一个

排列的 6 个位置中选出 4 个位置的子集, 有 $C(6, 4)=15$ 种方法. 把 b, a, a, a 以所要求的相对顺序放到这 4 个位置上, 只有 1 种方法. 接下来, 用 2 个 n 填充剩余的 2 个位置, 有 1 种方法. 所以最后的答案是 $15\times1\times1=15$ 种方法. □

练 习 5.3

练习概况 大部分问题都是本节例子的变形, 难度不大. 但是要小心区分排列和子集问题.

1. 投掷一枚骰子 5 次, 有多少种方法得到有 1 个 1、2 个 5 和 2 个 6 的结果序列?

2. 有多少种方法排列 MISSISSIPPI 中的字母?

3. **(a)** 由数字 3, 5, 7 可以形成多少个 7 位数?

(b) 问题 (a) 的 7 位数中有 3 个 3、2 个 5 和 2 个 7 的 7 位数的比例是多少?

4. 连续 6 天晚上邀请 3 个朋友之一参加晚宴, 有多少种方法可以做到没有人被邀请超过 3 次?

评价答案: **(a)** $6!/3!2!1!+6!/2!2!2!$; **(b)** $3^6-3\times C(6,4)\times 2^2$;

(c) $[C(6,1)+C(6,2)+C(6,3)]^3$.

5. 一堆硬币中有 1 分硬币、5 分硬币、10 分硬币和 25 分硬币, 有多少种方法从这堆硬币中挑出 8 枚硬币?

6. 如果投掷 3 枚相同的骰子, 可以记录多少不同的结果?

7. 有多少种方法从一群民主党人、共和党人和独立党人中选出由 15 位政治家组成的一个委员会, 使得在这个委员会中每个政党至少有两名成员? 另外, 如果任何党派的人士都不超过这个委员会成员的半数呢?

8. 9 个不同的人走进熟食店购买三明治. 3 个人总是购买金枪鱼三明治, 2 个人总是购买鸡肉三明治, 2 个人总是购买烤牛肉三明治, 有 2 个人购买这三种三明治中的任意一种.

(a)可能有多少种不同的三明治序列?

(b)可能有多少种不同的三明治的 (无序) 集合?

9. 有多少种方法从价值 1 美元的相同 1 分硬币、价值 1 美元的相同五分硬币、价值 1 美元的相同一角硬币中选出一个硬币序列, 如果:

(a) 总共选出了 8 枚硬币? **(b)** 总共选出了 15 枚硬币?

评价下面的答案: **(a)** $\binom{15+3-1}{15}$; **(b)** $\sum\limits_{k=0}^{10}\binom{15}{k}\binom{(15-k)+2-1}{(15-k)}$;

(c) $\binom{15+3-1}{15}-\sum\limits_{k=11}^{15}\binom{15}{k}\binom{(15-k)+2-1}{(15-k)}$.

10. **(a)** 把 5 个相同的苹果和 6 个相同的梨分给 3 个人, 使得每个人至少有 1 个梨, 有多少种方法?

(b) 把 5 个不同的苹果和 6 个不同的梨分给 3 个人, 使得每个人至少有 1 个梨, 有多少种方法?

11. 有一大堆水果, 其中有橙子、苹果、香蕉、桃和梨, 从这堆水果中选出 10 个水果且这 10 个水果正好包含两种不同的水果, 有多少种方法?

12. 有一大堆球, 其中有红球、白球和蓝球, 另外再加上 1 个粉球、1 个紫球和 1 个棕色球, 从这堆球中选出 8 个球有多少种方法?

13. 数字 1, 2, 2, 4, 4, 6, 6 的排列可以形成多少个大于 3 000 000 的数?

14. 函数 $f(x_1, x_2, \cdots, x_n)$ 有多少个 r 阶偏导数?

15. 正好包含 6 个不同数字的 8 数字序列有多少个?

16. 9 位数字中, 不同奇数的个数是不同偶数的个数的两倍的 9 位数字有多少个? (例如, 945 222 123 有奇数 9, 3, 5, 1, 有偶数 2, 4.)

17. 从 MISSISSIPPI 中选出的 5 个字母可以形成多少个可能的排列?

18. 在 1 到 25 之间 (包括两端) 可以重复地选出多少个 8 个数字的无序组? 其中这些数的和等于偶数的比例是多少?

19. REPETITION 的字母所形成的排列中, 第一个 E 出现在第一个 T 之前的排列有多少个?

20. 在一次国际田径赛中, 有 5 名美国选手、4 名俄罗斯选手、3 名法国选手和 1 名德国选手. 对于下面的条件, 这 13 名选手有多少种定级?

(a) 只计数国籍.

(b) 只计数国籍, 而且美国选手的最终成绩都在俄罗斯选手的前面.

21. MATHEMATICS 的字母所形成的排列中, 字母 TH 一起出现, 但是 TH 后面不紧跟着 E(不是 THE) 的排列有多少个?

22. PEPPERMILL 的字母所形成的排列中, 符合下面条件的排列有多少个?

(a) M 出现在所有元音的左边.

(b) 第一个 P 出现在第一个 L 的前面.

23. MISSISSIPPI 的字母所形成的排列中, 符合下面条件的排列有多少个?

(a) M 后面紧跟着一个 I.

(b) M 在 I 的旁边, 即 I 或者正好在 M 的前面或者正好在 M 的后面. 也可能一个 I 在 M 的前面且另一个 I 在 M 的后面, 要特别注意确保正确计数有模式 IMI 的排列.

24. 首先从 50 个人 (每一个人的身高都不同) 选出一个有 r 个人的子集, 接着再选出第二个有 s 个人的子集, 使得第一个子集中的每个人都比第二个子集中的每个人矮, 有多少种方法 (详细解释你的答案).

25. 有多少种方法把一组 $2n$ 个 α、$2n$ 个 β 和 $2n$ 个 γ 平分成两组 (分成各有 $3n$ 个字母的两组)? (注意, 两组之间是无序的; 没有第一个一半的说法.)

26. 有多少种方法在你右手 (拇指之外的)4 个手指头上放置 9 枚不同的戒指, 如果:

(a) 不考虑手指头上戒指的顺序?

(b) 考虑手指头上戒指的顺序? (提示: 需要技巧.)

27. 证明: $\sum P(10; k_1, k_2, k_3)=3^{10}$, 其中 k_1, k_2, k_3 是非负整数, 该式是对所有使得 $k_1+k_2+k_3=10$ 的可能三元组求和.

28. 6 个 0、5 个 1 和 4 个 2 所形成的排列中, 符合下面条件的排列有多少个?

(a) 第一个 0 在第一个 1 之前.

(b) 第一个 0 在第一个 1 之前, 且第一个 1 在第一个 2 之前.

29. 有 n 个类型的字母, 每种类型有 4 个字母. 这 $4n$ 个字母所形成的排列中, 每个字母都在一个相似字母的左右的排列有多少个?

30. 10 个人有多少种方法可以同时进行 5 次电话交谈?

31. 当投掷一枚硬币 n 次时, 下面各事件的概率是多少:

(a) 第一个正面正好在 m 个反面之后出现?

(b) 第 i 个正面正好在 m 个反面之后出现?

32. TINKERER 的字母所形成的排列中, 有 2 个连续元音但没有 3 个连续元音的排列有多少个?

33. 7 个 a、8 个 b、3 个 c 和 6 个 d 所形成的排列中, 没有连续的 ca 对或 cc 对出现的排列有多少个?

34. 5 个 α、5 个 β 和 5 个 γ 所形成的排列中, 在每一对连续的 α 之间至少有一个 β 且至少有一个 γ 的排列有多少个?

35. (a) 使用归纳法给出定理 1 的严格证明 (证明 1).

(b) 使用归纳法证明:

$$\binom{n}{r_1}\binom{n-r_1}{r_2}\binom{n-r_1-r_2}{r_3}\cdots\cdots\binom{n-r_1-r_2\cdots\cdots-r_{m-1}}{r_m}$$
$$=\frac{n!}{r_1!r_2!\cdots.r_m!}$$

5.4　分　配

一般来说, 分配问题等价于重复排列或重复选取的问题. 必须把特殊的分配问题分解成可以用简单的 (有或无重复的) 置换和组合来计数的子情况. 分配问题建模的一般方针是:

不同对象的分配等价于排列, 相同对象的分配等价于选取.

分配的基本模型

不同对象　把 r 个不同对象分配到 n 个不同盒子的过程等价于先把这些不同对象排成一行, 然后在每一个对象上贴上 n 个不同盒子名字中的一个. 盒子名字的最终序列是由 n 个项 (盒子名字) 形成的长度为 r 的重复排列. 因此有 $n\times n\times\cdots\times n(r个n)=n^r$ 种 r 个不同对象的分配. 如果 r_i 必须进入盒子 $i, 1\leqslant i\leqslant n$, 那么有 $P(r;r_1,r_2,\cdots,r_n)$ 种分配.

相同对象　把 r 个相同对象分配到 n 个不同盒子的过程等价于从 n 个盒子中选出一个 r 个盒子名字的无序子集. 因此有 $C(r+n-1,r)=(r+n-1)!/r!(n-1)!$ 种 r 个相同对象的分配.

例 1　派遣外交官

有多少种方法把 100 名外交官派遣到 5 个不同的大洲? 如果每个大洲必须分配 20 名外交官, 又有多少种派遣方法?

根据上面所讲的不同对象的分配模型, 这一派遣等价于在 5 个大洲目的地上排列 100 个人的长度为 100 的序列数量, 有 5^{100} 个这样的序列 (想象把手持贴有目地的的公文包的外交官排成一列). 每个大陆必须分配 20 名外交官的限制表明, 每个大洲的名字应该在这个序列中出现 20 次. 因此有 $P(100;20,20,20,20,20)=100!/(20!)^5$ 种方法. □

例 2　桥牌

在桥牌游戏中, 一副标准的 52 张纸牌随机分给玩家北、东、南和西各 13 张. 玩家西有所有 13 张黑桃的概率是多少? 每个玩家有一张 A 的概率是多少?

使用前面例子的推理, 把 52 张纸牌分配成 4 手不同的 13 张牌的分配方法有 $P(52;13,13,13,13)$ 种. 西家得到所有黑桃的分配数可以计数为把所有黑桃分配给西家的方法数, 只有 1 种方法, 再乘以把 39 张非黑桃牌分配成其他 3 手牌的方法数, 而这个方法数是 $P(39,13,13,13)$. 所以西家得到所有黑桃的概率是:

$$\frac{39!}{(13!)^3}\bigg/\frac{52!}{(13!)^4}=1\bigg/\frac{52!}{13!39!}=1\bigg/\binom{52}{13}$$

通过只考虑西家的牌 (忽视其他 3 手牌), 可以直接得到上面化简后的答案. 一次随机分配给西家 $C(52,13)$ 种可能中的一手 13 张牌. 因此, 唯一的一手 13 张黑桃有概率 $1/C(52,13)$.

为了计数每个玩家得到一张 A 的方法, 我们把符合条件的分配分成有 A 的分配和没有 A 的分配, 有 4! 方法在 4 个玩家中排列 4 张 A, 把其余的 48 张非 A 牌分配给每个玩家 12 张的方法数是 $P(48;12,12,12,12)$. 所以每个玩家得到一张 A 的概率是:

$$\frac{4!48!}{(12!)^4}\bigg/\frac{52!}{(13!)^4}=\frac{13!^4}{12!^4}\times\frac{4!48!}{52!}=13^4\bigg/\binom{52}{4}=0.105$$ □

例 3　分配糖果

有多少种方法把 20 根 (相同的) 红色棒棒糖和 15 根 (相同的) 黑色棒棒糖分给 5 个孩子?

使用相同对象的分配模型, 把 20 根相同的红色棒棒糖分给 5 个孩子的方法, 等价于从有 5 个不同名字的集合中重复地选出 20 个名字 (目的地) 的方法. 有 $C(20+5-1,20)$ =10 626 种方法. 根据相同的建模讨论, 15 根相同的黑色棒棒糖分配有 $C(15+5-1,15)$ =3 876 种分配方法. 红色和黑色棒棒糖的分配过程互不相关. 所以分配红色和黑色棒棒糖的方法数量是 10 626×3 876=41 186 376. □

例 4　分配相同对象和不同对象的组合

有多少种方法把 4 个相同的橙子和 6 个不同的苹果 (每个苹果的品种不同) 分到 5 个不同的盒子里? 在这些分配中, 每个盒子正好得到 2 个对象的分配的比例是多少?

有 $C(4+5-1,4)=70$ 种方法把 5 个相同的橙子分到 5 个不同的盒子里, 而且有 5^6=15 625 种方法把 6 个不同的苹果放入到 5 个不同的盒子里. 这两个过程是不相交的, 所以有 70×15 626=1 093 750 种方法把 4 个相同和 6 个不同的水果分配到 5 个不同的盒子里.

每个盒子里正好有 2 个水果的限制使问题复杂化. 为了计数不同对象的受限制分配, 必须精确地知道每个盒子里应该有多少个不同的对象. 但是在这个问题中,

可以进入一个盒子里的不同对象数依赖于在这个盒子里有多少相同的对象. 所以首先处理 (相同的) 橙子. 分配 4 个橙子有 3 种可能的策略 (在每一个盒子里不超过 2 个).

情况1　2 个盒子中的每个盒子里有 2 个 (相同的) 橙子, 而其余的 3 个盒子里没有橙子. 得到一对橙子的 2 个盒子可以有 $C(5,2)=10$ 种方法选出, 然后再把 6 个不同的苹果分配到其余 3 个盒子里, 一个盒子里有 2 个苹果, 有 $P(6;2,2,2)=90$ 种方法. 所以情况 1 有 10×90=900 种可能的分配.

情况2　一个盒子里有 2 个 (相同的) 橙子, 其他 2 个盒子中的每个盒子里有一个橙子, 剩余的 2 个盒子是空的. 有 2 个橙子的这个盒子可以有 $C(5,1)=5$ 种方法选出, 有一个橙子的 2 个盒子可以用 $C(4,2)=6$ 种方法选出 [或者把这两步结合在一起, 可以把这一分配想象成为把数字 2, 1, 1, 0, 0 放在 5 个盒子里的排列, 有 $P(5;1,2,2)=30$ 种方法]. 然后, 仍为空的 2 个盒子中的每个盒子都将得到 2 个苹果, 而且有 1 个橙子的 2 个盒子将各得到 1 个苹果. 因此, 6 个不同的苹果可以用 $P(6;2,2,1,1)=180$ 种方法分配, 所以总共有 5×6×180=5 400 种方法.

情况3　5 个盒子中的 4 个盒子里各有 1 个橙子, 有 $C(5,4)=5$ 种方法, 然后, 用 $P(6;2,1,1,1,1)=360$ 种方法分配苹果, 所以总共有 5×360=1 800 种方法.

对上面情况求和, 一共有 900+5 400+1 800=8 100 种分配使每个盒子里正好有 2 个对象. 这样的分配占 4 个橙子和 6 个苹果的总分配的比例是 8 100/1 093 750=0.007 4, 可能比你的猜测要小得多. (如果所有 10 个水果都不相同, 每个盒子里正好有 2 个水果的概率大约超过 50%. 为什么?) □

例 5　分配球

证明: 把 r 个相同的球分配到 n 个不同的盒子里, 且每个盒子里至少有 1 个球的方法数是 $C(r-1,n-1)$. 当第一个盒子里至少有 r_1 个球, 第二个盒子里至少有 r_2 个球, $\cdots\cdots$, 第 n 个盒子里至少有 r_n 个球时, 分配的方法数是 $C(r-r_1-r_2-\cdots-r_n+n-1,n-1)$.

每个盒子里至少有一个球的要求可以归类为与之等价的重复选取 (就如 5.3 节中的面包圈的选取一样). 可以为每个盒子准备一个标签, 放在一个盘子上, 然后用一张蜡纸盖在这些标签上, 然后用所有可能的方法挑出剩余的 $r-n$ 个标签, 这样就可以生成盒子的分配.

另外, 可以如下利用盒子直接处理这一限制. 首先在每个盒子里放入一个球, 然后在这些盒子里放入一个假底来把每个盒子里的球遮盖起来. 现在, 我们只剩下计数把剩余的 $r-n$ 个球无限制地放到 n 个盒子里的方法数.

无论使用哪种方法, 答案都是:

$$C((r-n)+n-1,(r-n))=\frac{[(r-n)+n-1]!}{(r-n)!(n-1)!}=C(r-1,n-1)\text{种方法}$$

对于第 i 个盒子里至少有 r_i 个球的情况, 我们首先在第 i 个盒子里放入 r_i 个球, 然后, 再用任意方法把剩余的 $r-r_1-r_2-\cdots-r_n$ 个球放入 n 个盒子里. 其方法数是:

$$\begin{aligned} & C((r-r_1-r_2-\cdots-r_n)+n-1,(r-r_1-r_2-\cdots-r_n)) \\ = \quad & C((r-r_1-r_2-\cdots-r_n)+n-1,n-1)\text{种方法} \end{aligned}$$ □

下面的计数问题在第 6 章生成函数的构建中起着重要的作用.

例 6 整数解

方程 $x_1+x_2+x_3+x_4=12$ 有多少个整数解, 其中 $x_i \geqslant 0$? 当 $x_i \geqslant 1$ 时有多少整数解? 当 $x_1 \geqslant 2$, $x_2 \geqslant 2$, $x_3 \geqslant 4$, $x_4 \geqslant 0$ 时, 有多少整数解?

说到这个方程的整数解, 我们的意思是指和等于 12 的 x_i 的整数值的有序集合, 例如, $x_1=2$, $x_2=3$, $x_3=3$, $x_4=4$. 可以把这样的问题建模成相同对象的分配问题, 或重复选取问题. 设 x_i 代表盒子 i 中的 (相同) 对象的数目, 或者代表所选择的类型 i 的对象数目. 这个方程右边的整数是要被分配或选取的对象数. 使用这两个模型中的任意一个, 我们看到整数解的数量是 $C(12+4-1,12)=455$ 种.

$x_i \geqslant 1$ 的解对应于在每个盒子里至少放入一个对象, 或每种类型至少选择一个对象. $x_1 \geqslant 2$, $x_2 \geqslant 2$, $x_3 \geqslant 4$, $x_4 \geqslant 0$ 的解对应于在第一个盒子里至少放入 2 个对象, 第二盒子里至少放入 2 个对象, 第三个盒子里至少放入 4 个对象, 第四个盒子里可以放入任意数量的对象 (或等价地在重复选择模型中考虑). 我们已经在例 5 中给出了这两种类型的分配问题的公式. 其答案分别是 $C(12-1,4-1)=165$ 和 $C((12-2-2-4)+4-1,4-1)=C(7,3)=35$. □

有整数值变量的方程称为*丢番图方程*(diophantine equation). 这是以希腊数学家丢番图 (Diophantus) 的名字命名的, 他在 2 250 年前就研究了它们.

现在, 我们已经看到了 3 个重复选取问题的等价形式.

重复选取的等价形式

(1) 从 n 个不同类型的对象中重复地选出 r 个对象的方法数.

(2) 把 r 个相同的对象分配到 n 个不同盒子里的方法数.

(3) $x_1+x_2+x_3+x_4=r$ 的非负整数解.

对读者来说, 能把用前面一种形式给出的问题转述成其他两种形式是很重要的. 很多学生发现版本 2 是寻找这一类问题的最方便的方法, 因为在一张纸上 (或者在脑子里) 画出分配是最容易的. 我们用于导出重复选择公式的关于热狗预定表的原始讨论, 实际上就是一个分配模型. 版本 3 是这一问题的最一般 (抽象) 形式. 它是我们在第 6 章中构建生成函数时所需的一种形式.

接下来, 我们给出 3 个问题, 并通过把问题转述为相同对象或不同对象的分配问题来解决它们.

例 7　巫医秘方原料

一位巫师去一家商店用 5 美元为他的妻子购买巫医秘方配制原料. 这家商店的蝙蝠尾巴的价格是每只 25 美分, 蜥蜴爪的价格是每只 25 美分, 蝾螈眼睛的价格是每只 25 美分, 小牛血的价格是每瓶 (1 品脱)1 美元. 这位巫师用他手中的钱能够采购不同的配料的组合数是多少?

第一步是以 25 美分作为金钱单位. 所以这位巫师有 20 个单位可以花费, 血的投资为 4 个单位, 而其他 3 项各要投资 1 个单位. 这个问题的整数解方程模型是

$$T+S+E+4B=20,\quad T,S,E,B\geqslant 0$$

由于 B 的系数为 4, 最简单的处理方法是首先明确指定要买多少血. 如果只买一品脱, B=1, 那么我们有 $T+S+E=16$, 而这个方程有 $C(16+3-1,16)=153$ 个非负整数解. 一般地, 如果 $B=i$, 那么有 $T+S+E=20-4i$, 这个方程有 $C((20-4i)+3-1,20-4i)=C(22-4i,2)$ 个解, 其中 $i=0$, 1, 2, 3, 4, 5. 对这些可能性求和, 我们得到

$$\binom{22}{2}+\binom{18}{2}+\binom{14}{2}+\binom{10}{2}+\binom{6}{2}+\binom{2}{2}=536\text{种不同的采购}$$ □

电子传感器的输出通常是由 0 和 1 构成的二进制序列. 当在某个传感器的输出中频繁地看到 0 和 1 的某个模式时, 那么很自然地会感觉到这个传感器可能有故障. 下面的例子要看一下二进制序列的特定模式的概率.

例 8　二进制模式

若干个 (多于 0 个)1, 后面跟着若干个 0, 再后面跟着若干个 1, 后面再跟着若干个 0 的长度为 10 的二进制序列的比例是多少? 一个这样的二元序列是 1110111000.

一共有 2^{10}=1 024 个 10 位二进制序列. 我们把这种 10 位二进制序列特定类型的计数问题如下建模为分配问题. 创建 4 个不同的盒子, 第一个盒子是为最初的一组 1 创建的, 第二个盒子是为后面的一组 0 准备的, 以此类推. 我们要把 10 个相同的标志 (称它们为 x) 分配到这 4 个盒子里. 因为每一个 0 或 1 的序列都必须是非空的, 所以每个盒子必须至少有 1 个标志. 根据例 5, 把 10 个 x 分配到 4 个盒子且没有空盒子的方法数是 $C(10-1,4-1)=84$. 所以, 总共有 84 个这样的二进制序列, 因此所有 10 位二进制序列中的 84/1 024≈8%是这种特殊类型的二进制序列. □

例 9　非连续元音

字母 a, e, i, o, u, x, x, x, x, x, x, x, x(8 个 x) 的排列中, 元音不连续的排列有多少个?

首先, 我们计数把这些 x 填充在 5 个元音位置之前、中间、之后的方法数. 我们要求至少有一个 x 在连续的元音之间. 想象一下 8 个 x 和 5 个 v 的排列. 可以这样对这一情况建模: 我们在 v 之前、中间和之后创建盒子, 然后再把这些 x 分配

到 6 个创建的盒子里且中间的 4 个盒子里至少有一个 x(以确保没有两个元音是连续的). 下面的模式说明了这些 x 的一种可能分配:

____v	x x _v	_x__v	_x__v	xxx_v	__x__
盒子1	盒子2	盒子3	盒子4	盒子5	盒子6

先在盒子 2, 3, 4, 5 中各放入一个 x. 然后, 把其余的 4 个 x 无限制地放入到这 6 个盒子里, 有 $C(4+6-1,4)=126$ 种方法. 这些方法计数了 x 和 v 的所有可能的模式 (v 的位置因我们对 x 的选择方法而被迫确定).

接下来, 对于每个模式, 在 5 个 v 的位置中排列 a, e, i, o, u, 有 5! =120 种排列方法. 总共有 120×126 = 15 120 种排列. □

最后用一张表来总结本章中所遇到的不同类型的基本计数问题.

从 n 个对象中排列、选取或分配 r 个对象的, 或到 n 个盒子的分配方法

	排列 (有序结果) 或 不同对象的分配	组合 (无序结果) 或 相同对象的分配
不重复	$P(n,r)$	$C(n,r)$
无限制重复	n^r	$C(n+r-1,r)$
限制重复	$P(n;r_1,r_2,\cdots r_m)$	—

练 习 5.4

练习概况 小心区分问题涉及的是不同对象还是相同对象. 练习 31~34 涉及问题重述 (不要求实际的数值答案). 练习 51 给出分析非连续问题的另一种重要方法.

1. 有多少种方法给 4 个孩子分配 36 块相同的软糖, 使得:
  (a) 没有限制? **(b)** 每个孩子可以得到 9 块? **(c)** 每个孩子至少得到 1 块?

2. 有多少种方法给 4 个孩子分配 16 个不同的玩具, 使得:
  (a) 没有限制?
  (b) 两个孩子各得到 6 个玩具且另两个孩子各得到 2 个玩具?
  (c) 每个孩子得到 4 个玩具?

3. 对于桥牌分牌, 下面事件的概率是:
  (a) 西家有 5 张黑桃、2 张红心、3 张方块和 3 张梅花?
  (b) 北家和南家都有 5 张黑桃, 西家有 2 张黑桃, 而东家有 1 张黑桃?
  (c) 一个玩家有所有的 A?
  (d) 所有的玩家的花色分布都是 (4, 3, 3, 3)?

4. 有多少种不同的方法给 3 个人分配 7 个 (相同的) 苹果、6 个相同的橙子和 7 个相同的梨, 使得:
  (a) 没有限制? **(b)** 每个人至少得到一个梨?

5. 有多少种方法给 4 位校长分配 16 个巧克力面包圈、10 个肉桂面包圈和 14 个糖粉面包圈, 使得对每一种面包圈每位校长至少得到 2 个?

6. 把 21 个不同对象分配到 3 个不同的盒子且一个盒子里的对象数是其他 2 个盒子中的对象数的 2 倍, 这样的分配有多少种?

7. VISITING 的字母的排列中, 没有连续 I 的排列有多少个?

8. 有多少种方法把 10 个相同的苹果和 5 个不同的橙子排成一行, 使得没有 2 个橙子是挨着出现的?

9. **(a)** COMBINATORICS 的字母的排列中, 没有连续元音出现的排列有多少个?

(b) 在 (a) 的排列中, 元音以字母顺序出现的排列又有多少个?

10. 排列字母表中的 26 个字母, 使得没有一对元音是连续出现的 (把 Y 认为是辅音), 有多少种方法?

11. 如果掷一枚硬币 20 次, 并得到 15 个正面和 5 个反面, 反面都不连续出现的概率是多少?

12. 方程 $x_1+x_2+x_3+x_4+x_5=28$ 有多少整数解, 其中:

(a) $x_i \geqslant 0$,　**(b)** $x_i > 0$,　**(c)** $x_i \geqslant i(i=1,2,3,4,5)$.

13. 方程 $x_1+x_2+x_3=0$ 有多少个整数解, 其中 $x_i \geqslant -5$?

14. 不等式 $x_1+x_2+x_3+x_4+x_5 < 100$ 有多少正整数解?

15. 有多少种方法把一副 52 张纸牌分成下面各大小的无序堆:

(a) 13 张牌的 4 堆?　**(b)** 8 张牌的 3 堆和 7 张牌的 4 堆?

16. 考虑把 10 本书分配给 3 个人, 使得每个人至少得到 1 本书的分配问题. 解释为什么下面的解决方案是错误的: 首先选出一本书给第 1 个人, 有 10 种方法; 然后, 再选出一本书给第 2 个人, 有 9 种方法; 再选出一本书给第 3 个人, 有 8 种方法; 最后有 7^3 方法来分配剩余的 7 本书.

17. 10 个雇员每人都带一份 (不同的) 礼物参加一次公司聚会. 每份礼物都分给由 Santa 随机选出的雇员 (一个雇员可以得到多份礼物). 至少有 2 个雇员没有得到礼物的概率是多少?

18. 有多少种方法把 k 个球分配到 n 个不同的盒子里 ($k < n$) 且每个盒子中至少有一个球, 如果:

(a) 这些球是不同的?　**(b)** 这些球是相同的?

19. 有多少种方法把 3 个不同的玩具熊和 9 个相同的棒棒糖分给 4 个孩子:

(a) 没有限制?　**(b)** 没有孩子得到 2 个或更多的玩具熊?

(c) 每个孩子得到 3 个 “东西”?

20. 假设要把 30 个不同的计算机游戏和 20 个不同的玩具分装到 3 个相同的装圣诞节礼物的袋子里. 第 1 个袋子里准备装入 20 个计算机游戏. 第 2 个袋子里准备装入 15 个玩具. 第 3 个袋子里准备装入 15 个礼物, 是游戏和玩具的任意混合. 有多少种方法把这 50 份礼物分装到这 3 个袋子里?

21. 假设投掷一枚硬币 12 次, 其中有 3 次正面和 9 次反面. 这样的序列中至少有 5 个连续反面的序列有多少个?

22. 满足下面条件的长度为 20 的二进制序列有多少个?

(a) 从若干个 0 开始, 即至少一个 0 的连续序列; 然后是若干个 1, 再然后是若干个 0, 然后再是若干个 1; 最后以若干个 0 结束.

(b) 重做问题 (a), 且附加每个 0 和 1 的连续序列的长度至少为 2 的限制?

23. 求满足下面条件的长度为 18 的二进制序列的个数: 以若干个 1 开始, 即 (至少) 一个 1 的连续序列, 然后是若干个 0, 再然后是若干个 1, 若干个 0, 而且有一个 1 的连续序列的长

度至少是 8.

24. EFFLORESCENCE 的字母的所有排列中, 有连续的 C、连续的 F 但没有连续的 E 的排列占总排列的比例是多少?

25. MISSISSIPPI 的所有排列中, 没有连续 S 的排列有多少个?

26. 把 15 个相同的对象分配到 4 个不同的盒子里, 且盒子 4 中的对象数是 3 的倍数, 那么有多少种分配方法?

27. 如果把 n 个不同的对象随机地分配到 n 个不同的盒子里, 那么下面各事件的概率是多少?

(a) 没有盒子是空的.　　**(b)** 正好有一个盒子是空的.

(c) 正好有两个盒子是空的.

28. 有多少种方法把 8 个球分配到 6 个盒子里且前 2 个盒子里总共至多有 4 个球, 如果:

(a) 球是相同的?　　**(b)** 球是不同的?

29. 在例 4 中, 如果所有 10 个水果都不相同, 则每个盒子里有 2 个水果的分配方法占总分配方法的几分之几? 为什么当所有水果不相同时, 这个分数会比较大?

30. **(a)** 有多少种方法从 1 952 个 1 分硬币、1 959 个 1 分硬币和 1 964 个 5 分硬币的 3 堆钱中取出 35 分钱?

(b) 从 1 952 个 1 分硬币、1 959 个 1 分硬币、1 964 个 5 分硬币和 1 971 个 25 分硬币的 4 堆钱中取出 35 分钱又有多少种方法呢?

31. 把下面的每一个排列问题描述成与其等价的分配问题:

(a) 从有若干个 a、若干个 b 和若干个 c 的一堆字母中选出 8 个字母的排列.

(b) 2 个 a、3 个 b 和 4 个 c 的排列.

(c) 从有若干个 a、若干个 b、若干个 c 和若干个 d 的一堆字母中, 选出 10 个字母且 a 和 b 的数量相同的排列.

(d) 从 2 个 a、3 个 b 和 4 个 c 中选出 4 个字母的排列.

32. 把下面的分配问题描述成等价的排列问题:

(a) n 个不同对象到 n 个不同盒子的分配.

(b) 把 15 个不同对象分配到 5 个不同盒子里且每个盒子有 3 个对象的分配.

(c) 把 15 个不同对象分配到 5 个不同盒子里且第 i 个盒子有 i 个对象 (其中 $i = 1, 2, 3, 4, 5$) 的分配.

(d) 把 12 个不同对象分配到 3 个不同盒子里且盒子 1 至多有 3 个对象、盒子 2 至多有 5 个对象的分配.

33. 把下面的选取问题描述成与其等价的分配问题和方程整数解问题:

(a) 从 31 种口味中选出 6 个冰激凌圆桶.

(b) 从有 5 个红色、4 个蓝色和 2 个粉色的一组弹球中选出 5 个弹球.

(c) 从 4 个品种的苹果中选出 12 个苹果, 且每个品种至少有 2 个.

(d) 从 4 个不同类型的夹心糖中选出 20 个夹心糖, 且每种类型有偶数个但不超过 8 个.

34. 把下面的分配问题描述成与其等价的选取问题和方程整数解问题:

(a) 把 30 个黑色筹码分装到 5 个不同的盒子里.

(b) 把 18 个红球分装到 6 个不同的盒子里, 且每个盒子里至少有 2 个球.

(c) 把 20 个马克笔分配到 4 个不同的盒子, 且第 1 个盒子和第 2 个盒子里的马克笔数目相同.

35. 如果有 3 名候选人和 30 名投票人, 那么有多少种可能的选举结果 (不同候选人的得票数)? 另外, 如果某个候选人得到过半数选票, 结果又如何?

36. 如果在一个班级中有 40 名学生和 5 名班级主席候选人, 那么在班级主席竞选中满足下面条件的选举结果有多少:

(a) 每名候选人至少得到 2 张选票?

(b) 某名候选人至多得到 1 张选票, 而其余人至少得到 2 张选票?

(c) 没有候选人得到过半数的选票?

(d) 正好有 3 名候选人并列得到最多的票数?

37. 0 到 10 000 之间有多少个数的各位数字之和符合下面条件?

(a) 等于 7? (b) 小于或等于 7? (c) 等于 13?

38. 不等式 $x_1+x_2+x_3+x_4 \leqslant 15$ 有多少个满足 $x_i \geqslant -10$ 的整数解?

39. 方程 $2x_1+2x_2+x_3+x_4=12$ 有多少非负整数解?

40. 一对方程 $x_1+x_2+\cdots+x_6=20$ 和 $x_1+x_2+x_3=7$ 有多少非负整数解?

41. 不等式 $x_1+x_2+\cdots+x_6 \leqslant 20$ 和 $x_1+x_2+x_3 \leqslant 7$ 有多少非负整数解?

42. 方程 $x_1+x_2+\cdots+x_5=20$ 有多少满足下面条件的非负整数解?

(a) 其中 $x_i \leqslant 10$. (b) $x_i \leqslant 8$. (c) $x_1 = 2x_2$.

43. 有多少种方法把 4 个红球、5 个蓝球和 7 个黑球分配到下面的盒子里:

(a) 没有限制地分配到两个盒子里?

(b) 分配到的两个盒子都不空?

44. UNIVERSALLY 的字母排列中, 使得 4 个元音以 2 簇 2 个连续元音的形式出现而且这 2 簇元音之间至少有 2 个辅音的排列有多少个? 这样排列的一个例子是 LN<u>UI</u>LYSV<u>AE</u>R.

45. UNIVERSALLY 的字母排列中, 使得没有 2 个元音连续出现而且辅音以字母表顺序出现的排列有多少个?

46. 由字母表中的 26 个字母所形成的 8 字母排列中, 至多包含 5 个元音中的 3 个元音, 且元音不连续的排列有多少个?

47. INSTITUTIONAL 的字母排列中, 同时具有下列所有性质的排列有多少个?

(a) 没有连续的 T, (b) 2 个 N 是连续的, (c) 元音按字母表中的顺序出现.

48. INSTRUCTOR 的字母排列中, 同时具有下列所有性质的排列有多少个?

(a) 元音以字母表的顺序出现, (b) 每对元音之间至少有 2 个辅音, (c) 以 2 个 T 开始或结束 (T 是连续的).

49. STATISTICS 的字母排列中, 同时具有下列所有性质的排列有多少个?

(a) 没有连续的 S, (b) 元音以字母表顺序出现, (c) 3 个 T 是连续的 (3 个 T 在一起).

50. (a) REVISITED 的字母排列中, 元音不以递增顺序出现, 即某个 I 在某个 E(或两个)E 的前面的排列有多少个?

(b) 没有连续的 E, 也没有连续的 I 的排列有多少个?

(c) 元音不以递增顺序出现, 且没有连续的 E 也没有连续的 I 的排列有多少个?

51. 本练习给出另一个计数某个类型的对象没有连续出现的排列的方法. 以练习 7 为例, 它是求 VISITING 的字母排列中没有连续的 I 出现的排列数. 排列 5 个辅音的方法数是 5!, 然后从 5 个辅音的前面、中间和后面的 6 个位置中不重复地选出 3 个位置的子集, 完成这一选择一共有 $C(6, 3)$ 种方法.

(a) 使用这一方法重做例 9.　　(b) 使用这一方法对练习 10 求解.

52. 在 WISCONSIN 的没有连续元音出现的字母排列中, W 与 I 相邻的排列的比例是多少?

53. 桥牌发牌时, 有多少种方法使南、北两家得到所有黑桃?

54. 桥牌发牌时, 每个玩家至少得到 3 张大牌的概率是多少 (大牌是 A、K、Q 和 J)?

55. 有多少种方法把 15 个橙子分配到 3 个不同的盒子且每个盒子里至多有 8 个橙子?

56. 对于下面的情况, 有多少种方法把 20 个玩具分给 m 个孩子, 使得前两个孩子得到相同数目的玩具?

(a) 这些玩具是相同的?　　(b) 这些玩具是不同的?

57. 有多少种方法在一次公司聚会上把 25 份不同的礼物分给 4 个人 (包括老板), 使得老板得到的礼物是第二重要人物得到的礼物的两倍?

58. 有多少种方法把 r 个相同的球分配到 n 个不同的盒子里, 使得正好有 m 个空盒子?

59. 从 $1, 2, \cdots, 20$ 中不重复地选出 6 个没有连续整数的整数子集 (例如, 如果 5 在这个子集中, 那么 4 和 6 就不能在其中), 有多少这样的子集?

60. 8 个 α、6 个 β 和 7 个 γ 所形成的排列中, 每个 α 都在另一个 α 的旁边 (在其至少一侧) 的排列有多少个? (提示: 关注相继的两簇 α.)

61. 使用 n 个 0 和 m 个 1 可以形成多少个有 k 串 0 的排列? [串是相同数字的 (1 个或更多) 连续集合; 例如, $\underline{000}111\underline{0}1\underline{00}$有 3 串 0(下划线部分).]

62. 一副 52 张纸牌的随机排列中正好有 k 串红心的概率是多少 (串的定义参见练习 61)?

63. 长度为 n 且模式 01 正好出现 m 次的二进制序列有多少个?

64. 有多少种方法把 20 面不同的旗挂到 12 根不同的旗杆上, 如果:

(a) 旗杆上旗的排列中, 从地面向上, 旗的顺序的不同使排列不同?

(b) 没有旗杆是空的, 且要考虑每根旗杆上旗的顺序?

65. 有多少种方法把 r 个相同的球分配到 n 个不同的盒子里, 使得前 m 个盒子总共至少包含 s 个球?

66. 有多少种方法给 13 个玩家每人分配 4 张纸牌, 使得正好有 11 个玩家每种花色有一张牌?

5.5　二项恒等式

本节说明为什么把数 $C(n,r)$ 称为二项式系数. 然后给出某些相关的二项式系数的等式. 我们使用 3 种技术来证明这些恒等式: 组合选取模型、“街区散步” 和二项式展开. 二项恒等式的研究本身是组合数学的重要分支, 而我们只了解一些粗浅的知识.

考虑多项式 $(a+x)^3$. 我们不是把 $(a+x)$ 与 $(a+x)$ 相乘, 然后再把积与 $(a+x)$ 相乘, 而是通过形式地逐项相乘, 把 3 个因子乘起来:

$$(a+x)(a+x)(a+x) = aaa + aax + axa + axx + xaa + xax + xxa + xxx$$

合并同类项, 我们把上式的右边化简成

$$a^3 + 3a^2x + 3ax^2 + x^3 \tag{1}$$

通过系统地把第一个因子中的项 a 或者 x 与第二个因子的项相乘, 再与第三个因子的项相乘得到所有积, 从而得到 $(a+x)^3$ 的形式展开式. 因为对这样的积中的每一项都有两种选择, 所以我们得到上面的 $2\times2\times2=8$ 个项的形式积. 如果要展开 $(a+x)^{10}$, 我们会得到 $2^{10}=1\,024$ 个项的形式积.

现在要问这样一个问题: $(a+x)^3$ 的展开式中包含 k 个 x 且包含 $(3-k)$ 个 a 的形式积有多少个? 这个问题等价于求 (1) 中的 $a^{3-k}x^k$ 的系数. 因为我们形成了所有可能的 a 和 x 的形式积, 又因为形式积正好是 a 和 x 的 3 字母序列, 所以我们只简单地要求有 k 个 x 和 $(3-k)$ 个 a 的所有 3 字母序列的数量. 这一答案是 $C(3,k)$, 所以 $(a+x)^3$ 的简化的展开式可以写成:

$$\binom{3}{0}a^3+\binom{3}{1}a^2x+\binom{3}{2}ax^2+\binom{3}{3}x^3$$

根据相同的推理, 我们看到在 $(a+x)^n$ 中 $a^{n-k}x^k$ 的系数等于 k 个 x 和 $(n-k)$ 个 a 形成的 n 字母序列的数量, 即 $C(n,k)$. 设 $a=1$, 我们有下面的定理.

二项式定理

$$(1+x)^n=\binom{n}{0}+\binom{n}{1}x+\binom{n}{2}x^2+\cdots+\binom{n}{k}x^k+\cdots+\binom{n}{n}x^n$$

练习中使用归纳法给出了这一展开的另一种证明. 与二项式定理同等重要的是, 我们已经建立了 n 个对象的 k 元子集的数量与 $(1+x)^n$ 中 x^k 的系数之间的等价性. 第 6 章将深入探索这一等价性.

现在考虑二项式系数的一些基本性质. 二项式系数的最重要的恒等式是对称恒等式:

$$\binom{n}{k}=\frac{n!}{k!(n-k)!}=\binom{n}{n-k} \tag{2}$$

简单地说, 这个等式说的是从 n 个对象的集合中选出 k 个对象的子集的方法数等于选出不在这个子集中的 $n-k$ 个对象的方法数.

另一个基本恒等式是

$$\binom{n}{k}=\binom{n-1}{k}+\binom{n-1}{k-1} \tag{3}$$

这个等式可以用代数方法证明. 这里, 我们给出它的组合讨论. 从 n 个人的集合中选出 k 个人组成委员会, 根据委员会中是否有指定的某人 P, 把所有 $C(n,k)$ 个可能的委员会分成两类. 如果 P 不是委员会成员, 那么有 $C(n-1,k)$ 种方法从其余 $n-1$ 个人中选出 k 个人组成委员会. 如果 P 是这个委员会的一员, 那么这一问题就简化成为从其余的 $n-1$ 人选出这个委员会的其余 $k-1$ 个成员. 有 $C(n-1,k-1)$ 种这样的方法. 因此, $C(n,k)=C(n-1,k)+C(n-1,k-1)$.

下面的例子给出另一个二项恒等式, 即可以用代数方法证明它, 也可以用组合讨论证明它.

例 1 证明

$$\binom{n}{k}\binom{k}{m}=\binom{n}{m}\binom{n-m}{k-m} \tag{4}$$

(4) 的左边计数了从 n 个人的集合中选出一组 k 个人的方法数, 然后再从这一组 k 个人之中选出 m 个领导者的子集的方法数. 等价地, 如右边所计数的那样, 可以先从 n 个人的集合中选出 m 个领导者, 然后从其余的 $n-m$ 个人中选出其余的 $k-m$ 个小组成员. 注意, 当 $m=1$ 时, (4) 的特殊形式是:

$$k\binom{n}{k}=n\binom{n-1}{k-1} \text{ 或 } \binom{n}{k}=\frac{n}{k}\binom{n-1}{k-1} \qquad \square(5)$$

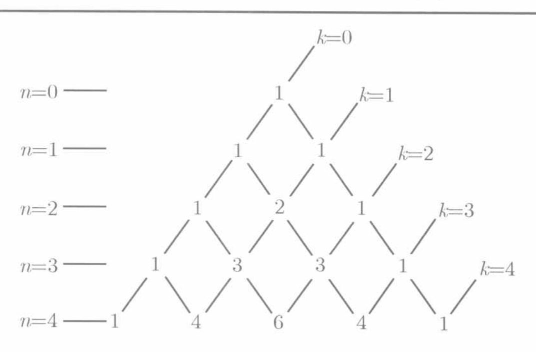
对于固定的整数 $n, k \leqslant n/2$ 时, 二项式系数 $C(n,k)$ 的值随着 k 的增大而增大. 然后, 当 $k \geqslant n/2+1$ 时, $C(n,k)$ 的值随着 k 的增大而减少. 为了证明这一点, 观察发现, 二项式系数增大, 当且仅当比 $C(n,k)/C(n,k-1)$ 大于 1.

$$\frac{\binom{n}{k}}{\binom{n}{k-1}}=\frac{\frac{n!}{k!(n-k)!}}{\frac{n!}{(k-1)!(n-k+1)!}}=\frac{n-k+1}{k}$$

所以, 当 $(n-k+1)/k>1$, 或等价地, 当 $n-k+1>k$ 时, $C(n,k)$ 递增. 使用 n 求解 k, 我们有 $k<(n+1)/2$. 这个不等式与 k 的整数值上界 $k \leqslant n/2$ 相同.

对所有非负整数 n, 使用 (3) 和恒等式 $C(n,0)=C(n,n)=1$, 可以递归地构建下面二项式系数的后继行, 我们称这个表格为帕斯卡三角形. 除每一行的第一个和最后一个数之外, 这一表格中的每个数都是前面一行的两个相邻数的和.

二项式系数表: 第 n 行的第 k 个数是 $\binom{n}{k}$



帕斯卡三角形有很好的组合解释, 这一解释是由 George Polya 给出的. 考虑一个人在图 5.1 所示的街区地图中遍历各街区的方法数. 这个人从这个网络的顶部开始, 这是标识为 (0, 0) 的地方, 沿着这个网络下行, 在每一个交叉口都要作出选择, 是向左走还是向右走 (为了方便起见, 设 "右" 就是你看这一页时的右边, 而不是向

下移动的这个人的右边). 用一对数 (n,k) 标签这个网络中的街角, 其中 n 表示从 (0, 0) 开始所经过的街区数, 而 k 表示在交叉口选择右边分支的次数. 图 5.1 给出从起点 (0, 0) 到街角 (6, 3) 的一条可能的路径.

到街角 (n,k) 的任意路径都可以写成从 (0, 0) 到 (n,k) 的路径上对随后街角做选择的一系列 (左或右) 分支. 这样的一个系列就是 k 个 R(右分支) 和 $(n-k)$ 个 L(左分支) 的序列. 为了沿图 5.1 所示的路径到达街角 (6, 3), 我们有分支序列 LLRRRL.

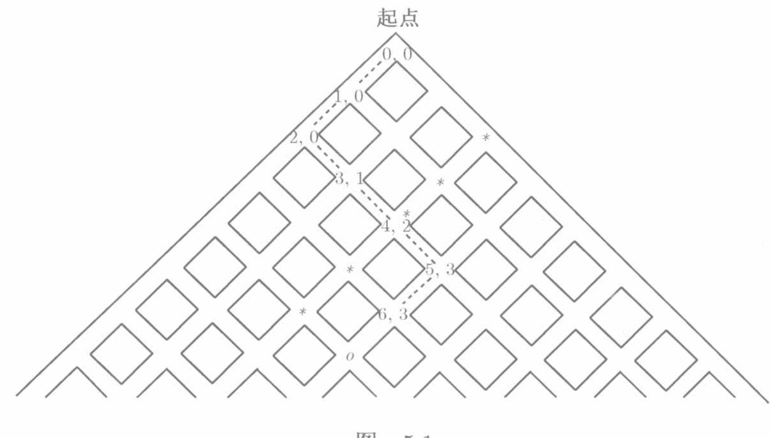


图 5.1

设 $s(n,k)$ 是从起点 (0, 0)(沿网络下行) 到街角 (n,k) 的可能路径数. 这个数目是 k 个 R 和 $(n-k)$ 个 L 的序列数, 所以 $s(n,k)=C(n,k)$. 二项式系数的另一个解释是委员会选举模型, 上面用这个模型证明了等式 (3) 和 (4).

下面说明如何使用二项式系数的 "街区散步" 模型得到恒等式 (3) 的另一个证明. 在从起始点 (0, 0) 到街角 (n,k) 的一条路线的末端, 一位沿街区散步的人从街角 $(n-1,k)$ 或者 $(n-1,k-1)$ 到达 (n,k). 例如, 为了到达图 5.1 的街角 (6, 3), 这个人或者到达 (5, 3), 然后取左分支到 (6, 3), 或者到达 (5, 2), 然后取右分支到达 (6, 3). 因此, 从 (0, 0) 到达 (n,k) 的路径数等于从 (0, 0) 到达 $(n-1,k)$ 的路径数加上从 (0, 0) 到 $(n-1,k-1)$ 的路径数. 于是我们证明了等式 (3): $s(n,k)=s(n-1,k)+s(n-1,k-1)$.

现在, 我们列出 7 个著名的二项恒等式, 并证明其中的 3 个 (其余留作练习). 这些二项恒等式都有实际意义. 包含二项式系数的和与积的展开式来自于复杂的真实世界的计数问题. 可以用这些恒等式简化这样的展开式.

$$\binom{n}{0}+\binom{n}{1}+\binom{n}{2}+\cdots+\binom{n}{n}=2^n \tag{6}$$

$$\binom{n}{0}+\binom{n+1}{1}+\binom{n+2}{2}+\cdots+\binom{n+r}{r}=\binom{n+r+1}{r} \tag{7}$$

$$\binom{r}{r}+\binom{r+1}{r}+\binom{r+2}{r}+\cdots+\binom{n}{r}=\binom{n+1}{r+1} \tag{8}$$

$$\binom{n}{0}^2+\binom{n}{1}^2+\binom{n}{2}^2+\cdots+\binom{n}{n}^2=\binom{2n}{n} \tag{9}$$

$$\sum_{k=0}^{r}\binom{m}{k}\binom{n}{r-k}=\binom{m+n}{r} \tag{10}$$

$$\sum_{k=0}^{m}\binom{m}{k}\binom{n}{r+k}=\binom{m+n}{m+r} \tag{11}$$

$$\sum_{k=n-s}^{m-r}\binom{m-k}{r}\binom{n+k}{s}=\binom{m+n+1}{r+s+1} \tag{12}$$

在此, 如果 $0\leqslant n<r$, 则 $C(n,r)=0$. 可以使用解释等式 (2), (3), (4) 时所使用的组合论证的"委员会"类型或者"街区散步论证"来解释这些等式. 我们给出等式 (6) 的委员会类型论证. 考虑下面计数 n 个人的集合的所有 (任意大小的) 子集的两种方法: (a) 求大小为 0, 1, 2 等的所有子集数的和, 这产生等式 (6) 的左边, (b) 通过检查第一个人是否在这个子集中, 第二个人是否在这个子集中等来计数所有子集, 于是有 $2\times2\times2\times\cdots\times2$($n$ 次)$=2^n$, 这是等式 (6) 的右边. 本节的最后给出 (6) 的一个更简单的证明.

一般地, 证明这样的恒等式的组合论证是由下面 3 个部分组成的: 指定由等式右边的项所计数的一个选取, 把这个选取分割成子情况, 并说明 (等式左边的) 和中的各项计数了这些子情况.

利用街区散步的证明的优点在于, 可以画出这个"证明"的图. 这个图展示了到达某个街角的所有路径, 这个量是等式的右边, 可以用到达某些中间街角的所有线路来分解这个量, 即等式左边的和.

例 2　利用街区散步和委员会选择的论证证明恒等式 (8).

作为这个恒等式的例子, 考虑 $r=2$ 和 $n=6$ 的情况. 在图 5.1 中, 街角 $(k, 2)$ 被标上一个星号 $*$, 其中 k =2,3,4,5, 6, 而街角 (7, 3) 被标上一个 o. 观察发现, 每一个带星号街角的右分支是从起点 (0, 0) 到街角 (7, 3) 的路径上的最后可能的右分支位置. 经过其中的一个右分支之后, 正好有一条路径继续到达街角 (7, 3), 即采用剩余分支的左分支. 一般地, 如果我们依据采用最后的右分支的街角, 把从 (0, 0) 到 $(n+1, r+1)$ 的所有路径分解成各个子情况, 我们得到恒等式 (8).

我们把街区散步模型重述为委员会选取: 如果第 k 次转向是向右, 这对应于选取委员会中的第 k 个人; 如果第 k 次转向是向左, 那么就不选取第 k 个人. 我们依据最后被选取的人, 把从 $n+1$ 个人中选出 $r+1$ 个委员会成员的方法分解成各种情况: 第 $(r+1)$ 个人, 第 $(r+2)$ 个人, $\cdots$, 第 $(n+1)$ 个人. 如果第 $(r+k+1)$ 个

人是最后被选取的, 那么有 $C(r+k, r)$ 种方法挑出这个委员会的前 r 个人. 恒等式 (8) 成立. □

例 3 通过街区散步的论证证明等式 (9).

从 (n,k) 到 $(2n,n)$ 的路径数等于从 $(0,0)$ 到 $(n,n-k)$ 的路径数, 因为这两种行程都经过了 n 个街区, 且有 $n-k$ 次向右 (k 次向左) 的转向. 所以从 $(0,0)$ 到 (n,k), 然后再到 $(2n,n)$ 的路径数等于 $C(n,k)\times C(n,n-k)$. 根据 (2), $C(n,n-k)=C(n,k)$, 因此, 从 $(0, 0)$ 经由 (n, k) 到 $(2n,n)$ 的路径数是 $C(n,k)^2$. 对所有 k 求和, 即, 对所有离开始点 n 个街区的中间街角求和, 我们计数了从 $(0, 0)$ 到 $(2n,n)$ 的所有路径. 所以, 这个和等于 $C(2n,n)$, 等式 (9) 成立. □

现在, 我们说明如何使用二项恒等式求其各项与二项式系数密切相关的和.

例 4 求和 $1\times 2\times 3+2\times 3\times 4+\cdots+(n-2)(n-1)n$.

这个和的一般项 $(k-2)(k-1)k$ 等于 $P(k,3)=k!/(k-3)!$. 回想一下, r 置换数与 r 组合数相差一个 $r!$ 的因子. 即 $C(k,3)=k!/(k-3)!3!=P(k,3)/3!$, 或 $P(k,3)=3!C(k,3)$. 所以上面给定的和可以重写成:

$$3!\binom{3}{3}+3!\binom{4}{3}+\cdots+3!\binom{n}{3}=3!\left(\binom{3}{3}+\binom{4}{3}+\cdots+\binom{n}{3}\right)$$

根据等式 (8), 这个和等于 $3!\binom{n+1}{4}$. □

例 5 求和 $1^2+2^2+3^3+\cdots+n^2$.

对于那些一般项不是 $C(n,k)$ 或 $P(n,k)$ 的乘积形式的和来说, 对它们求值的方法是把一般项代数分解成 $P(n,k)$ 类型的项的和. 对于上面这个问题, 一般项 k^2 可以写成 $k(k-1)+k$. 所以, 根据等式 (8), 上面给定的和可以重写为:

$$\begin{aligned}
&[(1\times 0)+1]+[(2\times 1)+2]+[(3\times 2)+3]+\cdots+[n(n-1)+n]\\
&=[(2\times 1)+(3\times 2)+\cdots+n(n-1)]+(1+2+3+\cdots+n)\\
&=\left(2\binom{2}{2}+2\binom{3}{2}+\cdots+2\binom{n}{2}\right)+\left(\binom{1}{1}+\binom{2}{1}+\cdots+\binom{n}{1}\right)\\
&=2\binom{n+1}{3}+\binom{n+1}{2}
\end{aligned}$$

注意, 作为例 5 的一部分, 我们证明了 $1+2+3+\cdots+n=C(n+1,2)=1/2n(n+1)$, 在附录 A.2 中的例 1 中, 我们使用归纳法给出它的证明. □

另外还有一种证明某些带有二项式系数的恒等式的简单方法. 我们从二项式定理的二项展开式开始. 然后用适当的值代替 x. 下面的恒等式就可以通过二项展开式得到: 通过设 $x=1$ 得到等式 (6); 通过设 $x=-1$ 得到下面的等式 (13); 通过对二项展开式的两边求导, 并设 $x=1$, 得到 (14).

$$(1+1)^n=2^n=\binom{n}{0}+\binom{n}{1}+\binom{n}{2}+\cdots+\binom{n}{n} \tag{6}$$

$$(1-1)^n = 0 = \binom{n}{0} - \binom{n}{1} + \binom{n}{2} - \cdots + (-1)^n \binom{n}{n} \tag{13}$$

或

$$\binom{n}{0} + \binom{n}{2} + \cdots = \binom{n}{1} + \binom{n}{3} + \cdots = 2^{n-1} \tag{13'}$$

(因为两端的和是 2^n)

$$n(1+x)^{n-1} = 1\binom{n}{1} + 2\binom{n}{2}x + 3\binom{n}{3}x^2 + \cdots + n\binom{n}{n}x^{n-1} \tag{14}$$

所以

$$n(1+1)^{n-1} = n2^{n-1} = 1\binom{n}{1} + 2\binom{n}{2} + 3\binom{n}{3} + \cdots + n\binom{n}{n} \tag{14'}$$

练 习 5.5

练习概况 组合恒等式是研究相当完善的一个领域, 本节只是粗浅地介绍了其中一些皮毛. 本练习后半部分的问题超出了本节所给出的例子的水平.

1. **(a)** 用代数方法证明等式 (3)(以阶乘的形式写出二项式系数).

(b) 用代数方法证明等式 (4).

(c) 证明 $C(n,k)/C(n,k-1) = (n-k+1)/k$.

2. 利用街区散步法证明下面的等式:

(a) 等式 (6); **(b)** 等式 (9); **(c)** 等式 (10); **(d)** 等式 (11).

3. 利用委员会选取举模型证明下面等式:

(a) 等式 (7); **(b)** 等式 (9); **(c)** 等式 (10); **(d)** 等式 (11); **(e)** 等式 (13).

4. 用数学归纳法证明下面的等式. [提示: 使用 (3).]

(a) 等式 (3); **(b)** 等式 (5); **(c)** 等式 (6); **(d)** 等式 (7); **(e)** 等式 (13′).

5. 用数学归纳法证明二项式定理.

6. 证明: 等式 (7) 可以作为 (11) 的特殊情况而得到.

7. 证明: $C(2n,n) + C(2n,n-1) = 1/2C(2n+2,n+1)$.

8. 如果 $C(n,3) + C(n+3-1,3) = P(n,3)$, 确定 n.

9. 用组合论证证明:

(a) $\binom{2n}{2} = 2\binom{n}{2} + n^2$;

(b) $(n-r)\binom{n+r-1}{r}\binom{n}{r} = n\binom{n+r-1}{2r}\binom{2r}{r}$.

10. 证明: $C(k+m+n,k)C(m+n,m) = (k+m+n)!/k!m!n!$

11. **(a)** 证明: $\binom{n}{1} + 6\binom{n}{2} + 6\binom{n}{3} = n^3$.

(b) 求值 $1^3+2^3+3^3+\cdots+n^3$.

12. **(a)** 求值 $\sum\limits_{k=0}^{n} 12(k+1)k(k-1)$. (提示: 使用例 4.)

(b) 求值 $\sum\limits_{k=0}^{n} (2+3k)^2$. **(c)** 求值 $\sum\limits_{k=0}^{n} k(n-k)$.

13. **(a)** 求和 $1 + 2\binom{n}{1} + \cdots + (k+1)\binom{n}{k} + \cdots + (n+1)\binom{n}{n}$, 方法是把这个和分成两个和, 每一个和都是本节中的等式.

(b) 求和 $\binom{n}{0}+2\binom{n}{1}+\binom{n}{2}+2\binom{n}{3}+\cdots$.

14. 在二项展开式 (或它的某个阶的导数, 等等) 中, 设 x 等于适当的值, 求:

(a) $\sum_{k=0}^{n}(-1)^k\binom{n}{k}$; (b) $\sum_{k=0}^{n}k(k-1)\binom{n}{k}$; (c) $\sum_{k=0}^{n}2^k\binom{n}{k}$;

(d) $\sum_{k=1}^{n}k3^k\binom{n}{k}$; (e) $\sum_{k=1}^{n}(-1)^k k\binom{n}{k}$;

(f) $\sum_{k=0}^{n}\frac{1}{k+1}\binom{n}{k}$; (g) $\sum_{k=0}^{n}(2k+1)\binom{n}{k}$.

15. 证明: $\sum_{k=m}^{n}\binom{k}{r}=\binom{n+1}{r+1}-\binom{m}{r+1}$.

16. 证明: $\sum_{k=1}^{n}\binom{m+k-1}{k}=\sum_{k=1}^{m}\binom{n+k-1}{k}$.

17. 证明: $\sum_{k=0}^{n-1}P(m+k,m)=\frac{P(m+n,m+1)}{(m+1)}$.

18. (a) 考虑一个出纳员旁边的一列 $2n$ 个不同人的序列. 假设有 n 个人从出纳员那里得到 1 美元, 而有 n 个人支付 1 美元. 证明: 这 $2n$ 个人的排列中, 使这位出纳员从不欠账 (即, 在每一个阶段, 支付 1 美元的人数至少与得到 1 美元的人数一样多) 的排列数等于 $\binom{2n}{n}-\binom{2n}{n+1}$. [提示: 利用街区散步模型的对称论证, 证明在某个阶段这位出纳员 (至少) 拖欠 1 美元的序列与有 $n+1$ 个人得到 1 美元的 $2n$ 个人的序列之间存在一一对应.]

(b) 重做 (a) 且附加下面的限制: 如果第一个人付 1 美元, 那么直到这一列人的最后一个人 (他得到 1 美元) 为止, 这位出纳员一直有正钱数的排列有多少个?

19. 证明: $\left[\binom{n}{0}+\binom{n}{1}+\binom{n}{2}+\cdots+\binom{n}{n}\right]^2=\sum_{k=0}^{2n}\binom{2n}{k}$.

20. 寻找使下式最大的 k 值:

(a) $\binom{n}{k}$; (b) $\binom{2n+k}{n}\binom{2n-k}{n}$.

21. 求值 $\sum_{k=1}^{n}\binom{n}{k}\binom{n}{k-1}$.

22. (a) 证明: $\binom{n}{1}+3\binom{n}{3}+5\binom{n}{5}+\cdots=2\binom{n}{2}+4\binom{n}{4}+6\binom{n}{6}+\cdots$.

(b) 上面等式每一侧的和是多少?

23. 求值 $\binom{n}{0}-2\binom{n}{1}+3\binom{n}{2}+\cdots+(-1)^n(n+1)\binom{n}{n}$.

24. 证明: $\sum_{k=0}^{n-1}(-1)^k\binom{n}{k+1}=1$. [提示: 把这个和重写成底部只有 k 的二项式系数.]

25. 证明: $\sum_{k=0}^{n}\frac{(2n)!}{k!^2(n-k)!^2}=\binom{2n}{n}^2$.

26. 用组合论证求值 $\sum_{k=0}^{n}\binom{n}{k}\binom{m}{k}$, 其中 $n\leqslant m$.

27. (a) 对于任意给定的 k, 证明: 整数 n 总可以表示成:

$$n=\binom{m_1}{1}+\binom{m_2}{2}+\cdots+\binom{m_k}{k},\qquad 1<m_1<m_2<\cdots<m_k.$$

(b) 证明: 对于给定的 k, 问题 (a) 中的表示是唯一的. [提示: 使用事实 $\binom{n}{k}-1=\binom{n-1}{k}+\binom{n-2}{k-1}+\cdots+\binom{n-k}{1}$.]

28. 证明: $\sum_{j=0}^{k}\binom{n+k-j-1}{k-j}\binom{m+j-1}{j}=\binom{n+m+k-1}{k}$.

29. 证明：$\sum\limits_{k=0}^{m}\frac{P(m,k)}{P(n,k)}=\frac{1}{\binom{n}{m}}\sum\limits_{k=0}^{m}\binom{n-k}{n-m}=\frac{n+1}{n-m+1}$, 其中 $m\leqslant n$.

30. 证明：$\sum\limits_{k=0}^{m}\frac{m!(n-k)!}{n!(m-k)!}=\frac{n+1}{n-m+1}$, 其中 $m\leqslant n$.

31. 如果 $C_{2n}=\frac{1}{n+1}\binom{2n}{n}$, 证明 $C_{2m}=\sum\limits_{k=1}^{m}C_{2k-2}C_{2m-2k}$.

32. 证明：$\sum\limits_{k=0}^{m}\binom{n}{k}\binom{n-k}{m-k}=2^m\binom{n}{m}$, 其中 $m<n$.

33. 考虑三维空间的街区散步问题. 利用组合论证证明：$P(n;i-1,j,k)+P(n;i,j-1,k)+P(n;i,j,k-1)=P(n+1,i,j,k)$.

34. 对下面的等式给出组合论证式的证明：

$$(x_1+x_2+\cdots+x_k)^n=\Sigma P(n;i_1,i_2,\cdots,i_k)x_1^{i_1}x_2^{i_2}\cdots x_k^{i_k}$$

其中, 这个和是对所有 $i_1+i_2+\cdots+i_k=n(i_j\geqslant 0)$ 进行求和的.

35. 对于所有的 $i_1+i_2+\cdots+i_k=n(i_j\geqslant 0)$ 求和, 证明：

(a) $\Sigma P(n;i_1,i_2,\cdots,i_k)=k^n$.

(b) $\Sigma i_1i_2\cdots i_kP(n;i_1,i_2,\cdots,i_k)=P(n,k)k^{n-k}$.

5.6　小结及参考文献

本章介绍了排列问题和组合问题所使用的基本公式和逻辑推理. 其中 4 个公式适用于有重复和没有重复的排列和选取问题, 而且还有 3 个法则适用于合成子问题. 但是这些公式和法则只为你解答大量的例子和练习搭建了基本框架而已. 在开始把这些问题分解成容易处理的子问题之前, 你需要对它们做全面彻底的逻辑分析. 这种关于可能性的逻辑分析源于计算机科学、概率和运筹学等领域. 它是离散数学的基本方法论, 也是学生们深入研究这门课程的重要技能. 注意, 这样的逻辑推理是数学建模的重要本质.

至少在 2 500 年之前, 人们就已经知道了排列的 $n!$ 公式 (有关组合数学历史的更详细内容, 请参见 David[2]). 在中国、印度和阿拉伯等国家 800 年前的著作中, 最初以原始方式提出了有关二项式系数和二项展开式的一些问题. 帕斯卡三角形出现在 14 世纪 Shih-Chieh[1] 的著作中 (参见原英文书封面). 西方最早的帕斯卡三角形出现在 200 年之后. 然而, 直到 17 世纪末, 才由 Jacob Bernoulli 给出了二项式定理的详细证明. 大约 1650 年左右由帕斯卡和费马提出的与赌博相关的概率的研究是现代组合数学的开始. 重复选取公式是下面的内容提出后不久发现的：投掷一枚硬币时, 在正好 r 个反面出现之后第 n 个正面出现的概率是 $\left(\frac{1}{2}\right)^{r+n}C(n+r-1,r)$($n$ 个正面的出现的前面和之间有 n 个放置 r 个反面的“盒子”). 重复排列公式是 1700 年左右由莱布尼茨在研究多项式定理的相关内容时发现的 (参见 5.5 节的练

① 朱世杰, 中国元代数学家, 对多元高次方程组解法、高阶等差级数求和、高次内插法都有深入研究, 主要著作有《算学启蒙》三卷和《四元玉鉴》三卷. —— 编者注

习 34). Jacques Bernoulli 的《猜度术》(1713 年) 是给出基本组合方法的第一本书. 第一本内容全面的关于置换和组合问题的教科书是由 Whitworth 于 1901 年写成的, 见文献 [3].

关于枚举的其他介绍性的书目列表, 可以参考本书末尾的参考文献.

1. R. Buck, *Advanced Calculus*, 3rd ed., Mc Graw-Hill, New York, 1978.
2. F. David, *Games, Gods, and Gambling: A History of Probability and Statistical Ideas,* Dover Press, New York, 1998.
3. W. Whitworth, *Choice and Chance*, 5th ed. (1901), Hafner Press, New York, 1965.

补充材料：第 5 章部分练习解答

5.1 节

13. **(a)** 所有 10 000 到 99 999 之间的数形成所有 5 位数. 答案是 $9\times10\times10\times10\times10$.

(b) 如果一个数是偶数, 那么其个位上的数字必须是偶数数字 (即, 0, 2, 4, 6, 8). 答案: $9\times10\times10\times10\times5$.

(c) 如果最左边的数字是 3, 那么有 9^4 个数. 如果任意其他数字是 3, 那么有 8×9^3 个数. 总共有 $9^4+4\times(8\times9^3)$ 个数.

(d) 回文数的最左边的数字和最右边的数字必须相同, 类似地对于第二位和倒数第二位的数字也必须相同. 于是这个问题就变成选出 3 个最左边数字的方法数 (由于回文的对称性, 2 个右边的数字根据其余数字而确定), 有 $9\times10\times10$ 种方法.

30. 对于 0 到 99 999 之间的数来回答这个问题会更加容易 (加上 0 并删去 100 000 并没有带来差异, 因为二者都不包含 5). 因此, 我们问在书写一个 5 位序列 (允许若干个 0 开头的 5 位数) 时, 5 出现了多少次. 考虑这样的问题, 在这些 5 位序列中, 5 在第 3 个位置上出现多少次, 即, 第 3 个位置是 5 的 5 位序列有多少个 5? 答案是 10^4. 对于 5 在这个序列中的所有位置, 答案是 5×10^4.

33. 一个数能够被 4 整除, 当且仅当它最右边的两位数字所组成的数能够被 4 整除. 另外, 为了能够被 4 整除, 个位数字 (最右边的数字) 必须是偶数, 在这个问题中, 这个数是 2 或者 4. 观察发现, 把任意给定的 10 位数与这两个偶数个位数合成, 例如 32, 34, 我们可以得到两个连续的偶数. 每两个连续的偶数之一正好能够被 4 整除. 所以为了生成数字 1, 2, 3, 4, 5 所形成的能够被 4 整除的数, 有 3 个位置上的数是任意的, 它们是千位、百位和十位, 个位是上面两个选择之一, 所以答案是 5^3.

5.2 节

45. 排列的总数是 $C(10,2)\times C(8,2)\times 6!$, 所以, 部分 (a) 和 (b) 的结果除以这个总数得到这些事件的概率.

(a) 用一个特殊的符号 V 取代 3 个连续元音序列. 现在要排列 8 个字母, C, N, R, R, S, T, T, V. 我们为 2 个 R 挑出一对位置 (大小为 2 的子集), 然后, 为 T 挑出一对 (剩余的) 位置, 然后再排列 4 个不同的字母, 有 $C(8,2)\times C(6,2)\times 4!$ 种排列. 接下来, 3 个元音的全排列有 3! 个排列来取代 V. 答案: $3!\times C(8,2)\times C(6,2)\times 5!$.

(b) 首先, 可以如部分 (a) 那样进行, 此时 V 是两个元音, 而 V′ 是单个元音. C, N, R, R, S, T, T, V, V′ 的所有排列是 $C'(9,2) \times C(7,2) \times 5!$. 对元音 V′ 有 3 种选择; 对 V, 剩余元音有 2 个排列. 总共有 $3{\times}2{\times}C(9,2) \times C(7,2) \times 5!$ 个排列. 然而, 如果 V 后面是 V′ 或者 V′ 后面是 V, 那么我们得到 3 个元音在一起. 3 个元音在一起的排列可以用两种方法生成, 例如, TIOUCRRTNS 来自于 T(IO)(U)CRRTNS 和 T(I)(OU)CRRTNS. 减去 3 个元音在一起的排列 [部分 (a)], 我们有$\{3{\times}2{\times}C(9,2) \times C(7,2) \times 5!\} - \{3! \times C(8,2) \times C(6,2) \times 4!\}$ 种排列.

51. (a) 如在大多数计数问题中一样, 这里的关键是关注需要选出的剩余数, 而不是已经知道在这个子集中的数. 为了具体些, 开始时假设 $k = 8$. 因为 8 是这个子集中最小的数, 所以这个子集中其他 3 个数都比 8 大, 选择的方法数是 $C(20-8,3)$. 对于一般的 k, 这个概率是 $C(20-k,3)/C(20,4)$.

(b) 如果 8 是这个子集中第二小的数, 那么存在一个更小的数, 有 $C(8-1,1)$ 种方法选出这个更小的数, 且有两个更大的数, 有 $C(20-8,2)$ 种方法选出这两个更大的数. 一般地, 概率是 $C(k-1,1) \times C(20-k,2)/C(20,4)$.

55. 本题中的 “技巧” 是计算没人有相同生日的概率, 然后再从 1 减去这个概率. 我们想要知道对 25 个人来说每个人的生日都不相同的可能生日的比例. 分母是 25 个人的所有可能的生日, 这个数字是 365^{25}. 分子是每个人有不同生日的所有可能, 这个数是 $P(365,25)$. 所以, 要求的概率是 $1-P(365,25)/365^{25}$. (使用计算器或计算机计算, 这个概率大约是 0.57.)

59. 解答技巧是确定 n 个不同对象中哪些对象将在这个集合中. 可以用 $C(n,0) + C(n,1) + C(n,2) + \cdots + C(n,n)$ 方法选出 n 个不同对象的任意子集, 这个集合中的剩余元素是由相同的对象组成的, 可以用 1 种方法确定剩余对象 (因为这些对象是相同的). 还有一个方法是, 我们可以选择使用或不使用每一个不同的对象, 最终答案是 2^n 种结果.

65. 分母是 $C(52,5)$.

(a) 选出一对 (相同点数的两张牌) 的方法有 $C(13,1) \times C(4,2)$ 种. 为了填充这一手牌中的其余牌, 选出第 2 种点数的一张牌, 有 48 种方法, 然后是第 3 种点数的一张牌, 44 种方法, 最后是第 4 种点数的一张牌, 有 40 种方法. 存在的问题是, 这一对剩余牌的计数是有序的, 即黑桃 7、红桃 Q 和方块 5 的选择序列与方块 5、红桃 Q 和黑桃 7 的序列是一样的. 上面的计数除以 3! 使得这个计数变成无序, 所以产生答案 $\{C(13,1) \times C(4,2) \times (48 \times 44 \times 40)/3!\}/C(52,5)$.

确定这手牌中其余牌的另一种方法是选出 3 张其他点数的子集, 有 $C(12,3)$ 种方法, 然后再从每一种点数中选出一张牌, 有 4^3 种方法, 于是结果是 $\{C(13,1) \times C(4,2) \times C(12,3) \times 4^3\}/C(52,5)$.

(b) 确定 5 张牌中每一张牌都是不同点数的概率, 然后用 1 减去这个概率: $1 - C(13,5) \times 4^5/C(52,5)$.

(c) 以特定顺序 (以递增或递减顺序) 分发任意一手 5 张牌的机会是 1/5!

(d) 选出在一手牌中出现的 5 种点数的一个子集, 有 $C(13,5)$ 种方法. 点数最低的 $k(1 \leqslant k \leqslant 4)$ 张牌必须是红心, 而其他牌是黑桃. 剩下的就是选择 k 的值, 有 4 种选择: $4{\times}C(13,5)$. (很有技巧性!)

73. (a) 凸 n 边形的每 4 个顶点形成唯一一个交点, 即连接这 4 个顶点的子集中的两个对立顶点形成的两条弦的交点. 因此, 答案是 $C(n,4)$.

(b) 弦的数量是 $C(n,2)-n$(除了这个多边形的 n 条边之外, 连接所有顶点对都可以形成弦). 每一个交点都把直线分成两段, 使线段数量增加 2. 于是, 部分 (b) 的答案是 $C(n,2)-n+2C(n,4)$.

(c) 边的总数 e 等于线段的数量加上这个多边形的边数, 即 $C(n,2)-n+2C(n,4)+n$. 于是 $e=C(n,2)+2C(n,4)$. 顶点数 v 是多边形的顶点数量加上交点, 所以有 $v=n+C(n,4)$. 根据欧拉公式, 区域数 (排除无限区域)$r=e-v+1=[C(n,2)+2C(n,4)]-[n+C(n,4)]+1=C(n,2)+C(n,4)-n+1$.

77. 我们要求 7 个 3 元子集的所有可能的无序组合, 使得这 7 个朋友中的每一对朋友正好在一个子集中. 考虑下面这样的一组 3 元子集表格 (X 表示该行的元素在该列的子集中):

朋友 \ 子集	1	2	3	4	5	6	7
A	X	X	X				
B	X			X	X		
C	X					X	X
D		X		X			X
E		X			X	X	
F			X	X		X	
G			X		X		X

设前 3 个 3 元子集是包含朋友 A 的那些子集. 有 $\left[\binom{6}{2}\times\binom{4}{2}\times\binom{2}{2}\right]/3!=15$ 个包含 A 的 3 元子集: 我们计数把其余 6 个朋友配对到这些包含 A 的 3 元子集中去的所有方法, 然后再除以 3! 使得这些配对是无序的. 设第一个子集既包含 B 也包含 A. 有 A 的 3 个 3 元子集的一般组合的形式是 (A, B, C′), (A, D′, E′), (A, F′, G′), 其中 C′ 是在有 A 和 B 的子集中的另一个朋友, 而 D′ 是不在第一个子集中的 (按字母表顺序的) 第一个朋友, E′ 是在有 A 和 D′ 的子集中的另一个朋友, F′ 的字母表顺序在 G′ 之前. 在上面的表格中, 用 C′, D′, E′, F′, G′ 取代 C, D, E, F, G, 唯一剩下的选择是 B 或 C′ 与 D′, F′ 和 E′, G′ 形成 3 元子集, 有 2 种选择. 所以有 15×2 种 7 个朋友的 7 个 3 元子集的组合, 使得每一对朋友在一个 3 元子集出现.

可以用 7! 种方法在连续 7 个晚间安排这样的 7 个 3 元子集的组合, 所以答案是 15×2×7!.

5.3 节

25. 计数从 3 种类型的字母中选出 $3n$ 个字母的方法, 然后减去一个字母出现 $2n+1$ 次或更多次的那些结果: $C(3n+3-1,3n)-3\times C((n-1)+3-1,n-1)$. 因为每一种分法都形成有 $3n$ 个字母的*两组*, 所以很显然我们应该把所有 $3n$ 字母组别的计数除以 2. 然而, 由每一种类型的 n 个字母组成的分组*没有*计数两次. 所以答案是 $1/2[C(3n+3-1,3n)-3\times C((n-1)+3-1,n-1)-1]+1$.

28. (a) 在 15 个位置上安排 4 个 2, 有 $C(15,4)$ 种方法; 然后把一个 0 放置在剩余位置的第一个位置, 有 1 种方法; 然后再为其余的 0 选出 5 个位置, 有 $C(10,5)$ 种方法. 答案: $C(15,4)\times C(10,5)$.

(b) 把一个 0 放置在第一个位置, 有 1 种方法; 然后为其余的 0 选出 5 个位置, 有 $C(14,5)$ 种方法. 把一个 1 放置在其余位置中的第一个位置, 有 1 种方法; 然后再为其余的 1

选出 4 个位置, 有 $C(8,4)$ 种方法. 所以答案: $C(14,5)\times C(8,4)$.

29. 不可能有正好相同的 3 个连续字母, 因为这样的话, 这个类型的第 4 个字母就落单了. 所以, 或者 2 个相同字母挨在一起, 或者 4 个相同字母挨在一起. 但是, 4 个字母挨在一起相当于 2 组挨在一起的 2 字母组挨在一起. 所以问题简化成计数对每个类型有 2 个字母的 n 个类型字母的 $2n$ 个字母的排列. 所以答案: $(2n)!/(2!)^n$.

31. **(a)** 总共有 2^n 个结果. 投掷序列以 m 个连续反面开头, 然后是一个正面. 可以用 $2^{n-(m+1)}$ 种方法完成这个序列. 概率是: $2^{n-(m+1)}/2^n=2^{-(m+1)}$.

(b) 如果第 i 个正面正好在 m 个反面后出现, 那么前 $m+(i-1)$ 次投掷包含 m 个反面和 $i-1$ 个正面, 有 $C(m+(i-1),m)$ 种方法. 其余的投掷没有限制, 有 $2^{n-(m+i)}$ 种方法. 概率: $C(m+(i-1),m)2^{n-(m+i)}/2^n=C(m+i-1,m)2^{-(m+i)}$.

34. 有 3 种情况:

(1) 在每一对 α 之间正好有一个 β 和一个 γ: 在 4 对 α 之间, 或者 β 在前或者 γ 在前, 有 2^4 种方法. 第 5 个 β 和第 5 个 γ 以及其余字母的序列可以认为是我们要排列的 3 个字母, 所以有 3! 种方法. 总共有 $2^4\times3!=96$ 种方法.

(2) 每一对 α 之间正好有一个 β, 且某一对 α 之间有两个 γ(或者是某一对 α 之间有两个 β 和每一对 α 之间正好有一个 γ): 有 4 种选择选出 2 个 γ 将插入的 α 对, 且排列其间的两个 γ 和一个 β 有 3 种方法. 在其他 3 对 α 之间, 则有 2^3 种选择 β 在前还是 γ 在前的方法, 第 5 个 β 进入这个排列的哪一端有 2 种选择. 因为有某对 α 之间有 2 个 β 的情况, 所以结果乘以 2, 我们得到 $2\times(4\times3\times2^3\times2)=384$ 种方法.

(3) 某对 α 之间有 2 个 β 且某对 α 之间有 2 个 γ: 这里又有 2 种子情况. 如果 2 个 β 和 2 个 γ 在同一对 α 之间, 那么有 4 种选择选出两个 β 和两个 γ 要插入的 α 对, 排列这对 α 之间的 β 和 γ 有 $C(4,2)$ 种方法, 而且对于其他 3 对 α 来说, 有 2^3 种方法使 β 在前或 γ 在前. 如果 2 个 β 和 2 个 γ 在不同对的 α 之间, 那么有 4×3 方法选出两个 β 和两个 γ 要插入的 α 对, 而且有 3^2 种方法排列两个 γ 和一个 β, 并排列一个 γ 和两个 β, 对于其他 2 对 α 之间, 有 2^2 种方法确定 β 在前还是 γ 在前. 总共有 $4\times C(4,2)\times2^3+4\times3\times3^2\times2^2=1\ 056$ 种方法.

综上所述, 3 种情况给出的总结果是: $96+384+1\ 056=1\ 536$ 种排列.

5.4 节

27. **(a)** $n!/n^n$.

(b) 挑出一个空盒子, 有 n 种选择; 然后挑出进入两个对象的盒子, 有 $n-1$ 种选择; 然后是进入这个盒子的两个对象, 有 $C(n,2)$ 种选择; 然后, 分配其余 $n-2$ 个对象进入其余 $n-2$ 个盒子, 一个对象进入一个盒子, 有 $(n-2)!$ 种方法. 概率是: $n\times(n-1)\times C(n,2)\times(n-2)!/n^n$.

(c) 挑出两个空盒子, 有 $C(n,2)$ 种选择. 分两种情况讨论: 情况 (i). 有一个盒子中有 3 个对象: 选择放入 3 个对象的盒子, 有 $n-2$ 种选择; 然后, 选择哪 3 个对象进入这个盒子, 有 $C(n,3)$ 种选择; 然后, 分配其余 $n-3$ 个对象进入其余的 $n-3$ 个盒子, 有 $(n-3)!$ 种方法. 情况 (ii): 有两个对象的盒子有两个: 选出有两个对象的两个盒子, 有 $C(n-2,2)$ 种选择; 然后, 选择哪两个对象进入这两个盒子中的第一个盒子, 哪两个对象进入第二个盒子, 有 $C(n,2)\times C(n-2,2)$ 种选择; 然后, 分配其余 $n-4$ 个对象进入其余 $n-4$ 个盒子, 有 $(n-4)!$ 种方法. 概率是: $C(n,2)\times\{(n-2)\times$

$C(n,3) \times (n-3)! + C(n-2,2) \times C(n,2) \times C(n-2,2) \times (n-4)!\}/n^n$.

28. 根据前两个盒子里有 0, 1, 2, 3, 4 个球分成 5 种情况.

(a) $\sum_{k=0}^{4} C(k+2-1,k) \times C(8-k+4-1,8-k)$.

(b) $\sum_{k=0}^{k} C(8,k) \times 2^k \times 4^{8-k}$：如果在前两个盒子里有 k 个不同的球, 那么从 8 个球中选出 k 个球, 有 $C(8,k)$ 种方法; 然后, 确定每一个球进入前两个盒子中的哪一个盒子, 有 2^k 种方法; 然后, 分配其余的 $8-k$ 个不同的球进入其余 4 个盒子, 有 4^{8-k} 种方法..

47. 首先, 我们处理连续 N 的限制, 方法是把连续的 N 贴合到一起形成一个合成字母. 接下来, 没有连续 T 的 3 个 T 和 9 个非 T 字母 (包含作为一个字母的连续 N) 的模式有 $C((9-2)+4-1,(9-2)) = C(10,7)$ 个. 剩下的子问题是在 9 个位置上排序 9 个非 T 字母. 有若干种完成这一工作的方法. 我们选择下面的策略. 首先, 把连续的 N 放入这个排列中, 有 9 种选择; 然后放入 L, 有 8 种选择; 然后放入 S, 有 7 种选择. 只有 1 种方法在剩余的位置上以字母表顺序放入 6 个元音. 答案是：$C(10,7) \times 9 \times 8 \times 7$.

52. 没有元音对的 WISCONSIN 的字母排列数：没有连续元音的元音和辅音的模式有 $C[(6-2)+4-1,(6-2)] = C(7,4)$ 个, 有 3 种方法在 3 个元音位置上分配 I, I, O; 有 6!/2!2! 种方法在辅音位置上分配辅音. 于是, 比例的分母是 $C(7,4) \times 3 \times 6!/2!2!$.

对于分子, 我们看 3 种情况：

(1) W 在第一个元音 I 旁边. 如果 W 正好在 I 之前 (把 W 和 I 贴合在一起), 那么, 保证没有连续元音的其他辅音的分配模式有 $C((5-2)+4-1,(5-2)) = C(6,3)$ 个; 而如果 W 在 I 的后面, 则有 $C((5-1)+4-1,(5-1)) = C(7,4)$ 种分配模式. 无论是哪种情况, 都有 2 种方法排序其他元音, 且有 5!/2!2! 种方法排列其余的辅音.

(2) 如果 W 在最后的元音 I 的旁边, 则结果与 (a) 相同.

(3) 如果 W 在中间的元音 I 旁边, 则有 $C((5-1)+4-1,(5-1)) = C(7,4)$ 种模式来分配其余的辅音, 以保证没有连续元音, 其中无论 W 在 I 之前还是之后：对于 W 有 2 种选择. 可以用 2×5!/2!2! 种方法放置其他元音和其他辅音. 我们必须减去有子序列 IWI 这样的排列 (即, W 是两个 I 之间的唯一辅音)：$C((5-1)+3-1,(5-1)) \times 2 \times 5!/2!2!$ 个排列. 因此, 分子是 $[2 \times C(6,3) + 4 \times C(7,4) - C(6,4)] \times 2 \times 5!/2!2!$.

59. 用 6 个 1 和 14 个 0 形成长度为 20 的二进制序列来表示哪一个整数在这个子集中 (在第 i 位置上的 1 表示 i 在这个子集中). 在这种形式中, 我们寻找有 6 个不连续的 1 的所有 20 位二进制序列. 答案是：$C((14-5)+7-1,(14-5))$.

61. 无论一串 0 有多长, 我们都用 R 来表示它. 然后, 排列 m 个 1 和 k 个 R, 使得没有连续的 R. 使用练习 59 中的推理, 有 $C(m+1,k)$ 个排列. 接下来, 选择每串 0 中 0 的个数. 我们要把 n 个 0 分配到 k 个盒子 (串), 且每个盒子里至少有一个 0, 有 $C(n-1,k-1)$ 种方法. 所以, 答案是 $C(m+1,k) \times C(n-1,k-1)$.

64. **(a)** 把旗杆首尾相连地放置在地面上, 旗杆之间画一道竖线 (|). 于是, 我们需要计数 20 面不同旗和 $(12-1)$ 个竖线 | 的所有排列, 所以答案是 31!/11! 种方法.

(b) 首先, 把 20 面相同的旗分配到 12 个旗杆上, 且没有旗杆是空的, 有 $C(20-1,12-1)$ 种方法. 现在, 如在 (a) 中一样, 把旗杆首尾相连地放置. 然后, 在 20 面相同旗子的位置上排列 20 面不同的旗, 有 20! 种方法, 所以答案是：$C(20-1,12-1) \times 20!$.

第 6 章　生 成 函 数

6.1　生成函数模型

本章介绍生成函数的概念. 我们研究生成函数来处理重复选取及排序问题中的特殊限制. 在第 7 章、第 8 章和第 9 章, 我们要用生成函数来解决其他组合问题. 生成函数是本书中所介绍的最抽象的问题解决技术之一. 但是一旦把它弄明白, 就可以把它们作为解决很多组合问题的最容易方法.

假设 a_r 是某个过程中选取 r 个对象的方法数. 那么 $g(x)$ 是关于 a_r 的**生成函数**(generating function), 如果 $g(x)$ 有下面的多项式展开:

$$g(x) = a_0 + a_1x + a_2x^2 + \cdots + a_rx^r + \cdots + a_nx^n$$

如果这个函数有无穷多项, 则称它为**幂级数**(power series).

在 5.5 节, 我们验证了著名的二项展开式:

$$(1+x)^n = 1 + \binom{n}{1}x + \binom{n}{2}x^2 + \cdots + \binom{n}{r}x^r + \cdots + \binom{n}{n}x^n$$

于是, $g(x) = (1+x)^n$ 是 $a_r = C(n,r)$ 的生成函数, 其中 a_r 是从 n 个对象的集合中选 r 个对象的子集的方法数. 回想一下, 我们首先通过考虑 $(a+x)^3$ 的形式积而得到 $(1+x)^n$ 的展开式:

$$\begin{aligned}(a+x)(a+x)(a+x) = &aaa + aax + axa + axx + xaa\\ &+xax + xxa + xxx\end{aligned}$$

当 $a=1$ 时, 我们得到:

$$\begin{aligned}(1+x)(1+x)(1+x) = &111 + 11x + 1x1 + 1xx + x11\\ &+x1x + xx1 + xxx \qquad (1)\end{aligned}$$

这样的形式展开式列出了第 1 个因子中的项乘以第 2 个因子的项, 再乘以第 3 个因子的项的所有形式. 确定 $(1+x)^3$ 中 x^r 的系数的问题, 进而更一般地 $(1+x)^n$ 中 x^r 的系数问题就可简化成计数有 r 个 x 和 $(n-r)$ 个 1 的不同形式积的数量. 这样的形式积是 r 个 x 和 $(n-r)$ 个 1 的所有序列. 所以 x^r 在 $(1+x)^3$ 中的系数是 $C(3,r)$, 而在 $(1+x)^n$ 中的系数是 $C(n,r)$.

这其中的关键是, 若干个多项式因子的乘积可以看成是把多项式每个因子的项乘到一起而生成的所有形式积的组合. 如果第 i 个多项式因子包含 r_i 个不同的项, 而且有 n 个因子, 那么将有 $r_1 \times r_2 \times r_3 \times \cdots \times r_n$ 个不同的形式积. 例如, 在 $(1+x)^n$ 的展开式中有 2^n 个形式积.

在 $(1+x+x^2)^4$ 的展开式中, 所有形式积的集合将是下面形式的序列:

$$\left\{\begin{matrix}1\\x\\x^2\end{matrix}\right\}\cdot\left\{\begin{matrix}1\\x\\x^2\end{matrix}\right\}\cdot\left\{\begin{matrix}1\\x\\x^2\end{matrix}\right\}\cdot\left\{\begin{matrix}1\\x\\x^2\end{matrix}\right\} \tag{2}$$

即, 积中每一项中都有 1 或者 x 或者 x^2, 例如, $x1x^2x$.

本章主要考虑 x 的幂的系数是 1 的多项式因子的乘积, 例如因子 $(1+x+x^2+x^3)$ 或者 $(1+x^2+x^4+x^6+\cdots)$. 这些因子完全由 x 的不同幂来刻画. 注意, $1=x^0$. 因此展开式 (1) 可以重写成:

$$\begin{aligned}(x^0+x^1)(x^0+x^1)(x^0+x^1)=&x^0x^0x^0+x^0x^0x^1+x^0x^1x^0+x^0x^1x^1+x^1x^0x^0\\&+x^1x^0x^1+x^1x^1x^0+x^1x^1x^1\end{aligned}$$

同样, 展开式 (2) 中的形式积可以重写成:

$$\left\{\begin{matrix}x^0\\x^1\\x^2\end{matrix}\right\}\cdot\left\{\begin{matrix}x^0\\x^1\\x^2\end{matrix}\right\}\cdot\left\{\begin{matrix}x^0\\x^1\\x^2\end{matrix}\right\}\cdot\left\{\begin{matrix}x^0\\x^1\\x^2\end{matrix}\right\} \quad \text{或} \quad x^{e_1}x^{e_2}x^{e_3}x^{e_4}, \text{其中} 0\leqslant e_i\leqslant 2 \tag{3}$$

把若干个这样的多项式因子相乘时, 可以用指数来重述确定 x^r 的系数的问题. 考虑 x^5 在 $(1+x+x^2)^4$ 的展开式中的系数. 这个系数是展开式 (3) 中指数和等于 5(如 $x^2x^0x^2x^1$) 的不同形式积. 可以把确定 x^5 在 $(1+x+x^2)^4$ 中的系数建模为方程整数解问题 (5.4 节的例 6). 等于 x^5 的形式积 $x^{e_1}x^{e_2}x^{e_3}x^{e_4}(0\leqslant e_i\leqslant 2)$ 的数量与下面方程的整数解的数目相同:

$$e_1+e_2+e_3+e_4=5, \quad \text{其中} 0\leqslant e_i\leqslant 2$$

根据 5.4 节讨论的重复组合的等价形式, 上面的方程整数解问题等价于从 4 种类型的集合中选出 5 个对象, 且每种类型至多有 2 个对象的问题. 它还等价于把 5 个相同的对象分配到 4 个不同盒子里且每个盒子里至多有 2 个对象的问题.

更一般地, x^r 在 $(1+x+x^2)^4$ 中的系数, 即等于 x^r 的形式积 $x^{e_1}x^{e_2}x^{e_3}x^{e_4}(0\leqslant e_i\leqslant 2)$ 的数量, 它等于下面方程的整数解的数量:

$$e_1+e_2+e_3+e_4=r, \quad \text{其中} 0\leqslant e_i\leqslant 2$$

而这一方程整数解的个数问题又等于从 4 个类型中选出 r 个对象且每种类型至多有 2 个对象的方法数量问题 (或把 r 个相同对象分配到 4 个盒子里且每个盒子里至多有 2 个对象的方法数问题). 因此, $(1+x+x^2)^4$ 是 a_r 的生成函数, 其中 a_r 是从 4 个类型对象中选出 r 个对象且每种类型至多有 2 个对象的方法数量 (或执行等价的分配).

在这一阶段, 我们只考虑如何为计数问题构建生成函数模型. 在 6.2 节, 我们将看到, 对生成函数实施各种代数操作后, 我们如何得到想要的系数.

我们已经说明了如何把 $(1+x+x^2)^4$ 的系数解释为某个重复组合问题, 或相同对象的分配问题的解. 这一推理线索也可以反过来: 给定某个重复组合或分配问题, 我们能够构建一个生成函数, 它的系数是这个问题的解. 现在给出如何构建这样的生成函数的例子. 我们使用方程整数解求解的中间模型, 来帮助构建生成函数.

例 1 寻找 a_r 的生成函数, 其中 a_r 是从 3 个绿球、3 个白球、3 个蓝球和 3 个金球中选出 r 个球的方法数.

可以用下面的方程的整数解数来对这一选取问题建模:

$$e_1+e_2+e_3+e_4=r, \quad 其中\ 0 \leqslant e_i \leqslant 3$$

这里的 e_1 代表所选取的绿球个数, e_2 代表白球个数, e_3 代表蓝球个数, e_4 代表金球个数. 为了更具体些, 假设 $r=6$. 于是整数方程模型是

$$e_1+e_2+e_3+e_4=6, \quad 其中\ 0 \leqslant e_i \leqslant 3$$

我们想要构建多项式因子的积, 使得当把这些因子乘开时, 可以得到所有形式积 $x^{e_1}x^{e_2}x^{e_3}x^{e_4}$, 其中每个指数 e_i 在 0 和 3 之间. 对于 $r=6$ 的情况, 可能的形式积之一是 $x^2x^0x^1x^3$. 于是, 我们需要 4 个因子, *每个因子应该包含选出各 e_i 的 x 幂的"目录"*. 即, 每个因子应该是 $(x^0+x^1+x^2+x^3)$. 所以, 要求的生成函数是 $(x^0+x^1+x^2+x^3)^4$, 或 $(1+x+x^2+x^3)^4$. □

例 2 为从 5 块巧克力、5 颗草莓、3 颗柠檬和 3 个樱桃面色圈中选出 r 个面色圈的方法数寻找生成函数. 另外, 在每种面包圈至少有一个的附加条件下, 重复这一例子.

可以用下面的方程整数解的个数来对第一个选取问题建模:

$$e_1+e_2+e_3+e_4=r, \quad 其中\ 0 \leqslant e_1, e_2 \leqslant 5,\ 0 \leqslant e_3, e_4 \leqslant 3$$

这里, e_1 代表选出的巧克力面包圈的数量, e_2 是草莓面包圈的数量, e_3 是柠檬面包圈的数量, e_4 是樱桃面包圈的数量. 我们想要构建一个多项因子的积, 使得当把这些多项因子相乘时, 我们能够得到所有形式积 $x^{e_1}x^{e_2}x^{e_3}x^{e_4}$, 使得每个 e_i 的边界由上面整数方程模型所确定. 对于 e_1 和 e_2, 因子是 $(x^0+x^1+x^2+x^3+x^4+x^5)$; 对于 e^3 和 e^4, 因子是 $(x^0+x^1+x^2+x^3)$. 于是, 所求的生成函数是 $(x^0+x^1+x^2+x^3+x^4+x^5)^2(x^0+x^1+x^2+x^3)^2$.

把每种类型至少有一个面包圈的附加限制加进来时, 整数解模型变成

$$e_1+e_2+e_3+e_4=r, \quad 其中 1 \leqslant e_1, e_2 \leqslant 5, \quad 1 \leqslant e_3, e_4 \leqslant 3$$

于是, e_1 和 e_2 的多项式因子变成 $(x^1+x^2+x^3+x^4+x^5)$, 而 e_3 和 e_4 的多项式因子变成 $(x^1+x^2+x^3)$. 所以, 要求的生成函数是 $(x^1+x^2+x^3+x^4+x^5)^2(x^1+x^2+x^3)^2$. □

例 3 使用生成函数模拟这样的计数问题: 从 3 种类型的对象中重复选出 6 个对象且每种类型至多有 4 个对象. 同时, 在无限制重复的情况下, 对这一问题建模.

可以用下面方程的整数解数量对这一选取问题建模:

$$e_1+e_2+e_3=6, \quad 其中 0\leqslant e_i\leqslant 4$$

这个问题不是要求选出 r 个对象的方法数的一般解. 然而, 在构建生成函数时, 我们自动地对 r 的所有值建模, 而不仅仅是 $r=6$. 想要得到对 6 个对象的这一问题的解意味着, 我们只对 x^6 的系数感兴趣, 即, 对等于 x^6 的形式积 $x^{e_1}x^{e_2}x^{e_3}$ 的数量感兴趣. 我们构建对每个 x^{e_i} 有 $(1+x+x^2+x^3+x^4)$ 的一个因子的生成函数. 所要的生成函数是 $(1+x+x^2+x^3+x^4)^3$, 我们想要知道 x^6 在其中的系数.

允许无限制重复的意思是每种类型的对象可以选任意多次, 所以任意指数值都是可能的 (尽管在这一特定问题中指数都不超过 6). 根据这一点, 答案是 x^6 在 $(1+x+x^2+x^3+\cdots)^3$ 中的系数, 其中 "$+\cdots$" 表明因子是无穷级数. □

对于上面无限制重复的情况, 可以使用生成函数 $(1+x+x^2+x^3+x^4+x^5+x^6)^3$, 因为我们只想要 x^6 的系数, 因此不使用 x 的更大幂. 然而, 在 6.2 节我们将看到, 在生成函数中使用无穷级数更加容易. 在重复选择问题中, 当我们要求从 3 种类型中无限制重复地选出 6 个对象时, 我们不附加任意类型至多 6 个的限制: 这一限制隐含在这个问题中, 没有必要明示. 同样的情况也适用于生成函数.

例 4 求 a_r 的生成函数, 其中 a_r 是把 r 个相同对象分配到 5 个不同盒子里的方法数, 其中前 2 个盒子里的对象个数是不超过 10 的偶数, 而其他盒子里的对象个数是 3 个到 5 个.

尽管这个例子中的限制是特意设计的, 但是我们仍然可以很容易地把这一问题模拟成带有适当限制的方程整数解问题. 我们要计数下面方程的所有整数解:

$$e_1+e_2+e_3+e_4+e_5=r, \quad 其中 e_1 和 e_2 是偶数, 0\leqslant e_1,e_2\leqslant 10, 且 3\leqslant e_3,e_4\leqslant 5$$

在各 e_i 的给定限制之下, 为了得到所有形式积 $x^{e_1}x^{e_2}x^{e_3}x^{e_4}x^{e_5}$, 我们需要有 5 个因子的积, 每个因子包含满足 x^{e_i} 要求的 x 的幂的目录. 例如, x^{e_1} 的目录是 $(1+x^2+x^4+x^6+x^8+x^{10})$. 所求的生成函数是 $g(x)=(1+x^2+x^4+x^6+x^8+x^{10})^2(x^3+x^4+x^5)^3$. □

经过一些实践, 构造生成函数模型就会变得容易一些. 然而, 生成函数的思想却远不是这样简单. 在第 5 章中, 我们解决了类似的选取和分配问题, 这些问题都包含与二项式系数相关的明确公式. 现在, 我们要把这些问题模拟成某个系数问题, 这个系数代表某些多项式因子乘积中的形式积的个数. 我们要写下来的就是这些多项式因子.

我们无法为某个特定量 (例如 6 个对象) 的选取问题或分配问题设计生成函数. 它必须模拟所有可能的对象数量的问题. 生成函数把关于这些子问题的所有必要信息都存储于一个函数之中.

在 6.2 节, 我们将把确定生成函数中特定系数的工作简化成为一系列的机械式代数操作. 这与函数是模拟复杂的选取问题, 还是模拟复杂的分配问题无关. 组合推理只用于构建生成函数的过程中.

练 习 6.1

练习概况 前 21 道练习都是简单的生成函数建模. 剩下的问题是更具挑战性的建模问题. 练习 25～29 使用多元生成函数.

1. 对于下面各表达式, 列出其中指数和等于 4 的所有形式积的集合.

 (a) $(1+x+x^3)^2(1+x)^2$

 (b) $(1+x+x^2+x^3+x^4)^2$

 (c) $(1+x^3+x^4)^2(1+x+x^2)^2$

 (d) $(1+x+x^2+x^3+\cdots)^3$

2. 构建 a_r 的生成函数, 其中 a_r 是下面等式的整数解的个数:

 (a) $e_1+e_2+e_3+e_4+e_5=r$, 其中 $0\leqslant e_i\leqslant 4$.

 (b) $e_1+e_2+e_3=r$, 其中 $0<e_i<5$.

 (c) $e_1+e_2+e_3+e_4=r$, 其中 $2\leqslant e_i\leqslant 6$, e_1 为偶数, e_2 为奇数.

 (d) $e_1+e_2+e_3+e_4=r$, 其中 $0\leqslant e_i$.

 (e) $e_1+e_2+e_3+e_4=r$, 其中 $0<e_i$, e_2 和 e_4 为奇数, 且 $e_4\leqslant 3$.

3. 构建 a_r 的生成函数, 其中 a_r 是从下面的对象中选出 r 个对象的方法数:

 (a) 5 个红球、4 个黑球和 4 个白球.

 (b) 5 个果冻、4 根甘草糖、8 个棒棒糖, 且每种糖果至少有 1 个.

 (c) 无限量的 1 美分硬币、5 美分硬币、10 美分硬币和 25 美分硬币.

 (d) 有 7 种灯泡, 且第一种和第二种灯泡必须是奇数个.

4. 构建 a_r 的生成函数, 其中 a_r 是 r 个相同对象到下面盒子里的分配数:

 (a) 5 个不同的盒子, 且每个盒子里至多有 5 个对象.

 (b) 4 个不同的盒子, 且每个盒子里的对象数是 3 个到 6 个.

 (c) 7 个不同的盒子, 且每个盒子里至少有 1 个对象.

 (d) 3 个不同的盒子, 且第一个盒子里至多有 5 个对象.

5. 使用生成函数模拟字母 M, A, T, H 的 6 组合数, 其中 M 和 A 可以出现任意多次, 而 T 和 H 至多只能出现一次. 我们想要的是这个生成函数中的哪个系数?

6. 使用生成函数求解: 有 5 种不同类型的热狗时, 从中选出 r 个热狗的不同组合数是多少?

7. 使用生成函数求: 把 18 个巧克力兔子分配到 4 个复活节篮子里, 且每个篮子里至少有 3 个兔子的分配方法数. 我们想要的是哪个系数?

8. **(a)** 使用生成函数模拟有 27 名学生给 4 名候选人投票时, 班级主席选举中不同的选举结果数. 我们想要的是哪个系数?

 (b) 假设作为候选人的每名学生都投他们自己的票, 那么此时的生成函数和要求的系数各是什么?

 (c) 假设没有候选人得到过半数的选票, 重做问题 (a).

9. 寻找 a_r 的生成函数, a_r 是重复地从 n 个对象的集合中选出 r 个对象的组合数.

10. 假定从 u 种类型的糖果中的每一种拿出 1 块, 从 v 种类型糖果中的每一种拿出 2 块, 从 w 种类型的糖果中的每一种拿出 3 块, 寻找选出 r 个糖果的方法数的生成函数.

11. 求 a_k 的生成函数, 其中 a_k 是 n 种类型对象的 k 组合数, 其中第一种类型有偶数个, 第二种类型有奇数个, 其他类型有任意多个.

12. 求 a_r 的生成函数, 其中 a_r 是下面的分配数: 把 r 个相同的对象分配到 q 个不同的盒子里且第一个盒子里的对象数是 r_1 到 s_1 之间的一个奇数, 第二个盒子里的对象数是 r_2 到 s_2 之间的一个偶数, 其他盒子里至多有 3 个对象. 注意, 假设 r_1 和 s_1 是奇数, r_2 和 s_2 是偶数.

13. 求 a_r 的生成函数, 其中 a_r 是由 n 枚不同的骰子组合出的点数和为 r 的方法数.

14. 求 a_r 的生成函数, 其中 a_r 是投掷 6 枚不同骰子一次得到点数和为 r 的方法数, 如果:

(a) 前 3 枚骰子是奇数, 后 3 枚骰子是偶数;

(b) 第 i 枚骰子的点数不是 i.

15. 构建 a_r 的生成函数, 其中 a_r 是方程 $e_1+e_2+e_3+e_4=r$ 的整数解个数, 其中 $-3 \leqslant e_j \leqslant 3$.

16. 求 0 到 999 999 之间的数字和等于 r 的整数个数的生成函数.

17. 求从 8 种口味的口香糖中选出 r 片的组合数的生成函数, 其中每种口味都是 5 片装.

18. **(a)** 使用生成函数对这样的一个问题建模: 20 个相同的球分配到 5 个不同的盒子里, 且每个盒子里的球数是 2 到 7 个, 求这个分配数.

(b) 从问题 (a) 的每一个多项式因子中提取一个 x^2. 用组合方法解释这个修改后的生成函数.

19. 使用生成函数对从 $1,2,\cdots,n$ 中选出 5 个整数且没有两个整数连续的方法数建模. 当 $n=20$ 时, 我们想要的是哪个系数? 对于一般的 n, 情况又如何?

20. 解释为什么 $(1+x+x^2+\cdots+x^r)^4$ 不是 a_r 的正确生成函数, 其中 a_r 是重复地从 4 种类型对象中选出 r 个对象的方法数. 正确的生成函数是什么?

21. 解释为什么 $(1+x+x^2+x^3+x^4)^r$ 不是 a_r 的正确生成函数, 其中 a_r 是把 r 个果冻分配给 r 个孩子且任何孩子得到的果冻都不超过 4 个.

22. 证明: 方程 $e_1+e_2+e_3+e_4=r$, $0 \leqslant e_1 \leqslant e_2 \leqslant e_3 \leqslant e_4$ 的整数解数的生成函数是:

$$(1+x+x^2+\cdots)(1+x^2+x^4+\cdots)(1+x^3+x^6+\cdots)(1+x^4+x^8+\cdots)$$

23. 求把 r 美分兑换成 1 美分、5 美分和 10 美分的方法数的生成函数.

24. 在一次全国歌唱比赛中, 每个州选出 5 名选手. 使用生成函数对选出 20 名参加半决赛选手的方法数建模, 如果

(a) 每个州至多有 1 个人; **(b)** 每个州至多有 3 个人.

25. 求生成函数 $g(x,y)$, 其中 $x^r y^s$ 的系数是把 r 块巧克力和 s 块棒棒糖分配给 5 个孩子, 使得任何孩子得到的棒棒糖数都不超过 3 个的方法数.

26. **(a)** 求生成函数 $g(x,y,z)$, 它的 $x^r y^s z^t$ 的系数是把 r 个红球、s 个蓝球和 t 个绿球分配给 n 个人, 且分给每个人的每种球的数量是 3 到 6 之间时的分配数.

(b) 还假设每个人得到的红球至少与蓝球一样多.

(c) 还假设前 3 个人得到红球和蓝球的数量相同.

(d) 还假设任何人得到绿球和红球的数量都不相同.

27. 求生成函数 $g(x,y,z)$, 它的 $x^r y^s z^t$ 的系数是 8 个人从一大盆苹果、橙子和香焦中选出 r 个苹果、s 个橙子和 t 个香焦的方法数.

28. 寻找生成函数 $g(x_1,x_2,\cdots,x_m)$, 它的 $x_1^{r_1}x_2^{r_2}\cdots x_m^{r_m}$ 的系数是 n 个人选出 r_1 把类型 1 的椅子, r_2 把类型 2 的椅子, $\cdots$, r_m 把类型 m 的椅子的方法数, 如果:

(a) 每个人挑出一把椅子;

(b) 每个人或者挑出一种类型的 2 把椅子, 或者根本不挑椅子;

(c) 人 i 只从一种类型中挑出至多 r 把椅子.

29. 如果 $g(x_1, x_2, \cdots, x_p) = (x_1 + x_2 + \cdots + x_p)^n, p > n, g(x_1, x_2, \cdots, x_p)$ 的展开式中有多少 x_i 的指数都不大于 1 的项? 这些项之一的系数是什么?

6.2 计算生成函数的系数

现在, 我们开发计算生成函数系数的一些代数技巧. 所有这些方法的宗旨都是把复杂的生成函数化简成简单的二项式类型的生成函数, 或若干个二项式类型生成函数的积. 为了方便引用, 表 6.1 列出了本节使用的所有多项式恒等式和展开式.

表 6.1 多项式展开式

(1) $\frac{1-x^{m+1}}{1-x} = 1 + x + x^2 + \cdots + x^m$

(2) $\frac{1}{1-x} = 1 + x + x^2 + \cdots$

(3) $(1+x)^n = 1 + \binom{n}{1}x + \binom{n}{2}x^2 + \cdots + \binom{n}{r}x^r + \cdots + \binom{n}{n}x^n$

(4) $(1-x^m)^n = 1 - \binom{n}{1}x^m + \binom{n}{2}x^{2m} + \cdots + (-1)^k\binom{n}{k}x^{km} + \cdots + (-1)^n\binom{n}{n}x^{nm}$

(5) $\frac{1}{(1-x)^n} = 1 + \binom{1+n-1}{1}x + \binom{2+n-1}{2}x^2 + \cdots + \binom{r+n-1}{r}x^r + \cdots$

(6) 若 $h(x) = f(x)g(x)$, 其中 $f(x) = a_0 + a_1x + a_2x^2 + \cdots$, $g(x) = b_0 + b_1x + b_2x^2 + \cdots$, 则

$$h(x) = a_0b_0 + (a_1b_0 + a_0b_1)x + (a_2b_0 + a_1b_1 + a_0b_2)x^2 + \cdots$$
$$+ (a_rb_0 + a_{r-1}b_1 + a_{r-2}b_2 + \cdots + a_0b_r)x^r + \cdots$$

(6) 中的生成函数乘法规则就是多项式乘法的标准公式. 等式 (1) 可以用多项式的 "长除法" 验证. 我们重述这一等式, 把 (1) 的两边同时乘以 $(1-x)$, 得到 $(1-x^{m+1}) = (1-x)(1+x+x^2+\cdots+x^m)$. 我们通过长乘法证明右边的积是 $1-x^{m+1}$.

$$\begin{array}{l} 1 + x + x^2 + \cdots + x^m \\ \underline{1 - x \qquad\qquad\qquad\qquad} \\ \ \ 1 + x + x^2 + \cdots + x^m \\ \underline{\ \ -x - x^2 - x^3 \cdots - x^m - x^{m+1}} \\ 1 \qquad\qquad\qquad\qquad\quad - x^{m+1} \end{array} \qquad (*)$$

如果 m 是无限大, 使得 $1+x+x^2+\cdots+x^m$ 变成无穷级数 $1+x+x^2+\cdots$, 那么乘法过程 $(*)$ 将产生一个幂级数, 其中每个 x^k 的系数是 $0(k>0)$[读者可以自己对 $(*)$ 验证这一事实]. 我们得出结论: $(1-x)(1+x+x^2+\ldots) = 1$. [从分析的角度看, 这个等式对 $|x| < 1$ 成立; 随着 m 趋向无限大, $(*)$ 中的剩余项 x^{m+1} 趋向 0.] 这个等式的两边除以 $(1-x)$, 我们得到等式 (2).

我们已经在 6.1 节的开始对二项展开式 (3) 做了解释. 展开式 (4) 是通过设 $y = -x^m$, 并利用等式 (3) 展开 $(1+y)^n$ 得到的:

$$[1 + (-x^m)]^n = 1 + \binom{n}{1}(-x^m) + \binom{n}{2}(-x^m)^2 + \cdots + \binom{n}{k}(-x^m)^k + \cdots + \binom{n}{n}(-x^m)^n$$

根据恒等式 (2), $(1-x)^{-n}$, 或等价的 $\left(\frac{1}{1-x}\right)^n$, 等于

$$(1+x+x^2+x^3+\cdots)^n \tag{7}$$

通过计算指数和等于 r 的那些形式积的个数来确定 x^r 在 (7) 中的系数. 如果 e_i 代表形式积中第 i 项的指数, 那么指数和等于 r 的形式积 $x^{e_1}x^{e_2}x^{e_3}\cdots x^{e_n}$ 的个数等于下面方程的整数解个数:

$$e_1+e_2+e_3+\cdots+e_n=r, \text{其中} e_i \geqslant 0$$

在 5.4 节的例 5 中, 我们已经证明这个方程的非负整数解个数是 $C(r+n-1,r)$. 因此, x^r 在 (7) 中的系数是 $C(r+n-1,r)$. 这就证明了展开式 (5).

使用公式 (1) 到 (6), 可以确定各种生成函数的系数: 首先, 执行代数操作把给定的生成函数简化成为 $(1+x)^n$, $(1-x^m)^n$, $(1-x)^{-n}$ 之一, 或者两个这样的展开式的积; 然后使用展开式 (3) 到 (5) 和乘法规则 (6) 来得到任意想要的系数. 我们用下面的例子说明若干常用的化简方法.

例 1 求 x^{16} 在 $(x^2+x^3+x^4+\cdots)^5$ 中的系数. x^r 的系数是多少?

为了化简这一展开式, 从每一个多项式因子中提取 x^2, 然后运用恒等式 (2).

$$\begin{aligned}(x^2+x^3+x^4+\cdots)^5 =&[x^2(1+x+x^2+\cdots)]^5\\ =&x^{10}(1+x+x^2+\cdots)^5 = x^{10}\frac{1}{(1-x)^5}\end{aligned}$$

因此, x^{16} 在 $(x^2+x^3+x^4+\cdots)^5$ 中的系数就是 x^{16} 在 $x^{10}(1-x)^{-5}$ 中的系数. 但是 x^{16} 在后面这个展开式中的系数应该等于 x^6 在 $(1-x)^{-5}$ 中的系数. [即 $(1-x)^{-5}$ 中的 x^6 项与 x^{10} 相乘得到 $x^{10}(1-x)^{-5}$ 中的 x^{16} 项.] 根据展开式 (5), x^6 在 $(1-x)^{-5}$ 的系数是 $C(6+5-1,6)$.

更一般地, x^r 在 $x^{10}(1-x)^{-5}$ 中的系数等于 x^{r-10} 在 $(1-x)^{-5}$ 中的系数, 即 $C((r-10)+5-1,(r-10))$. □

观察发现, $(x^2+x^3+x^4+\cdots)^5$ 是 a^r 的生成函数, 其中 a_r 是从 5 种类型的对象中重复地选出 r 个对象且每种类型至少有 2 个对象的方法数. 第 5 章如下解决了这样的问题: 首先在每种类型中挑出 2 个对象, 有 1 种方法; 然后计算挑出其余 $r-10$ 个对象, 有 $C((r-10)+5-1,(r-10))$ 种方法. 在例 1 的生成函数分析中, 我们做了代数处理: 从每个因子中提取 x^2, 总共提取 x^{10}, 然后寻找 x^{r-10} 在 $(1+x+x^2+\cdots)^5$ 中的系数, 这是从 5 种类型对象中无限制重复地选出 $r-10$ 个对象的方法数的生成函数.

从每个因子中提取 x 的最高公共幂这一典型代数技巧, 对应于在相关的选取问题的求解中所使用的 "技巧". 这样的对应是我们使用生成函数的一个主要原因: 代数技术自动地为我们做组合推理.

例 2 从 20 个人中募集 15 美元, 并假设前 19 个人中每个人都可以捐出 1 美元 (或者不捐), 而第 20 个人或者捐 1 美元或者捐 5 美元 (或者不捐). 利用生成函数求实现这一募捐的方法数.

这种募集问题等价于求方程 $x_1+x_2+\cdots+x_{19}+x_{20}=15$ 的整数解的数量, 其中 $x_i=0$ 或 1, $i=1,2,\cdots,19$ 且 $x_{20}=0,1,5$. 这一方程整数解问题的生成函数是 $(1+x)^{19}(1+x+x^5)$. 我们要求的是 x^{15} 的系数. 这一生成函数的第一部分有二项展开式:

$$(1+x)^{19}=1+\binom{19}{1}x+\binom{19}{2}x^2+\cdots+\binom{19}{r}x^r+\cdots+\binom{19}{19}x^{19}$$

如果设 $f(x)=(1+x)^{19}$, 而设 $g(x)=1+x+x^5$, 那么可以使用 (6) 计算 x^{15} 在 $h(x)=f(x)g(x)$ 中的系数. 设 a_r 是 x^r 在 $f(x)$ 中的系数, b_r 是 x^r 在 $g(x)$ 中的系数. 我们知道 $a_r=\binom{19}{r}$, 而且 $b_0=b_1=b_5=1$(其他的 b_i 等于 0).

这样, 根据 (6), x^{15} 在 $h(x)=f(x)g(x)$ 中的系数是

$$a_{15}b_0+a_{14}b_1+a_{13}b_2+\cdots+a_0b_{15}$$

因为 b_0, b_1 和 b_5 是 $g(x)$ 中仅有的非 0 系数, 上面的代数式可以化简成

$$a_{15}b_0+a_{14}b_1+a_{10}b_5$$

替换等式 (8) 中的各 a 和 b 的值, 我们有

$$\binom{19}{15}\times1+\binom{19}{14}\times1+\binom{19}{10}\times1=\binom{19}{15}+\binom{19}{14}+\binom{19}{10}$$ □

例 3 有多少种方法把 25 个相同的球分配到 7 个不同的盒子里, 使得第一个盒子里的球数不超过 10, 而且其他 6 个盒子里可以有任意多个球?

使用等式 (1) 和 (2), 把 r 个球分配到 7 个盒子且第一个盒子里的球数至多是 10 的分配数的生成函数是:

$$\begin{aligned}&(1+x+x^2+\cdots+x^{10})(1+x+x^2+\cdots)^6\\&=\left(\frac{1-x^{11}}{1-x}\right)\left(\frac{1}{1-x}\right)^6=(1-x^{11})\left(\frac{1}{1-x}\right)^7\end{aligned}$$

设 $f(x)=1-x^{11}, g(x)=(1-x)^{-7}$. 使用展开式 (5), 我们有

$$\begin{aligned}g(x)=&(1-x)^{-7}=1+\binom{1+7-1}{1}x+\binom{2+7-1}{2}x^2\\&+\cdots+\binom{r+7-1}{r}x^r+\cdots\end{aligned}$$

我们想要的是 x^{25}(要分配 25 个球) 在 $h(x)=f(x)g(x)$ 中的系数. 如在例 2 中一样, 只需考虑在两个多项式 $(1-x^{11})$ 和 $(\frac{1}{1-x})^7$ 的积中能够产生 x^{25} 项的项. $f(x)$ 中仅有的非 0 系数是 $a_0=1$ 和 $a_{11}=-1$. 所以, x^{25} 在 $f(x)g(x)$ 中的系数是:

$$a_0b_{25}+a_{11}b_{14}=1\times\binom{25+7-1}{25}+(-1)\times\binom{14+7-1}{14}$$

例 3 的答案的组合解释是, 我们计数把 25 个球无限制地分配到 7 个盒子里的方法数, 有 $C(25+7-1,25)$ 种方法, 然后再减去违反第一个盒子的限制的分配数, 即减去第一个盒子里至少有 11 个球的分配数, 有 $C((25-11)+7-1,(25-11))$ 种分配 (首先把 11 个球放入到第一个盒子里, 然后随意地分配剩余的球). 同样, 生成函数自动地执行了这样的组合推理. □

下面的例子使用前 3 个例子中使用的全部技巧, 来解决第 5 章中的组合方法不能解决的一个问题.

例 4 有多少种方法从 7 种类型的玩具中选出 25 个玩具且每种玩具的个数在 2 到 6 之间?

设 a_r 是从 7 种类型的玩具中选出 r 个玩具且每种玩具的个数在 2 到 6 之间的方法数, a_r 的生成函数是:

$$(x^2+x^3+x^4+x^5+x^6)^7$$

我们想要的是 x^{25} 的系数. 如在例 1 中一样, 我们从每个因子中提取 x^2, 得到

$$[x^2(1+x+x^2+x^3+x^4)]^7 = x^{14}(1+x+x^2+x^3+x^4)^7$$

现在把原来的问题化简成寻找 $x^{25-14}=x^{11}$ 在 $(1+x+x^2+x^3+x^4)^7$ 中的系数. 使用恒等式 (1), 可以把这个生成函数重写成:

$$(1+x+x^2+x^3+x^4)^7 = \left(\frac{1-x^5}{1-x}\right)^7 = (1-x^5)^7\left(\frac{1}{1-x}\right)^7$$

设 $f(x)=(1-x^5)^7$, $g(x)=(1-x)^{-7}$. 根据展开式 (4) 和 (5), 我们有

$$f(x) = (1-x^5)^7 = 1-\binom{7}{1}x^5+\binom{7}{2}x^{10}-\binom{7}{3}x^{15}+\cdots$$

$$g(x) = \left(\frac{1}{1-x}\right)^7 = 1-\binom{1+7-1}{1}x+\binom{2+7-1}{2}x^2+\cdots$$
$$+\binom{r+7-1}{r}x^r+\cdots$$

为了求 x^{11} 的系数, 只需考虑这两个多项式 $(1-x^5)^7$ 和 $(\frac{1}{1-x})^7$ 的积中产生 x^{11} 项的项. 在 $f(x)=(1-x^5)^7$ 中下标小于等于 11 的非零系数只有 a_0, a_5, a_{10}[参见上面的 $f(x)$ 的展开式]. 包含这 3 个系数且产生 x^{11} 项的积是:

$$\begin{array}{ccccc} a_0b_{11} & + & a_5b_6 & + & a_{10}b_1 \end{array}$$
$$= 1\times\binom{11+7-1}{11}+\left(-\binom{7}{1}\right)\times\binom{6+7-1}{6}+\binom{7}{2}\times\binom{1+7-1}{1}$$ □

我们对例 4 的最终结果给出如下组合解释. 第一项 $C(11+7-1,11)$ 计数了从 7 种类型的玩具中无限制地选出 11 个玩具的方法数, 其中 11 是在我们首先从每种类型挑出 2 个玩具之后要选出的玩具数. 下一项 $-7C(6+7-1,6)$ 是减去至少从某种

类型中又选出了 5 个玩具这样的不符合限制的 7 种情况 (我们从某种类型中选出 5 个玩具, 然后再从 7 种类型中选出 11−5=6 个玩具). 最后一项 $C(7,2)C(1+7-1,1)$ 是加回所有 $C(7,2)$ 种同时在两种类型违反了限制的情况, 即, 从两种类型中都至少选出 5 个对象的情况 [我们从两种类型中各选出 5 个对象, 然后再从 7 种类型中再选出 $11-(2\times 5)=1$ 个玩具]. 我们将在第 8 章探讨这种组合方法所蕴含的逻辑.

本节的最后, 我们讨论如何利用生成函数证明二项恒等式. 我们把二项恒等式的右边表示成某个生成函数 $h(x)$ 的特殊项的系数, 而把左边表示成生成函数 $f(x)$ 和 $g(x)$ 的积中的相同项的系数, 其中 $h(x)=f(x)g(x)$.

例 5　二项恒等式

证明二项恒等式

$$\binom{n}{0}^2+\binom{n}{1}^2+\cdots+\binom{n}{n}^2=\binom{2n}{n}$$

这个等式的右边是 x^n 在 $(1+x)^{2n}$ 中的系数. 左边的项都是 $(1+x)^n$ 的系数. 我们想要的生成函数的积是 $(1+x)^n(1+x)^n$. 即, 设 $f(x)=g(x)=(1+x)^n$, 使得 $f(x)g(x)=h(x)=(1+x)^{2n}$. 这时, $a_r=b_r=C(n,r)$. 根据乘法规则 (6), x^n 在 $f(x)g(x)$ 中的系数是

$$a_0b_n+a_1b_{n-1}+\cdots+a_nb_0$$

$$=\binom{n}{0}\binom{n}{n}+\binom{n}{1}\binom{n}{n-1}+\cdots+\binom{n}{0}\binom{n}{n}$$

$$=\binom{n}{0}^2+\binom{n}{1}^2+\cdots+\binom{n}{n}^2\quad\left[\text{因为}\binom{n}{r}=\binom{n}{n-r}\right]$$

等于 x^n 在 $(1+x)^n(1+x)^n$ 中的系数, 因此, 我们得到要求的等式. □

希望读者把刚才给出的组合恒等式的生成函数证明与 5.5 节例 3 中相同恒等式的组合证明进行比较.

练　习　6.2

练习概况　前 29 道练习与本节中的系数计算的例子相似. 练习 31~36 是关于二项式等式的证明问题 (其中有几个问题需要相当的技巧). 练习 38~42 和 44 介绍概率生成函数的话题 (对概率有所了解的话将是很有帮助的).

1. 求 x^9 在 $(1+x+x^2+x^3+\cdots)^n$ 中的系数.

2. 求 x^r 在 $(x^5+x^6+x^7+\cdots)^7$ 中的系数.

3. 求 x^8 在 $(1+x^2+x^4)(1+x)^m$ 中的系数.

4. 求 x^{18} 在 $(x+x^2+x^3+x^4+x^5)(x^2+x^3+x^4+\cdots)^5$ 中的系数.

5. 求 x^{17} 在 $(x^2+x^3+x^4+x^5+x^6+x^7)^4$ 中的系数.

6. 求 x^{52} 在 $(x^{10}+x^{11}+\cdots+x^{25})(x+x^2+\cdots+x^{15})(x^{20}+\cdots+x^{45})$ 中的系数.

7. 求 x^{34} 在 $(x^3+x^4+x^5+x^6+x^7)^7$ 中的系数.

8. 求 x^{24} 在 $(x+x^2+x^3+x^4+x^5)^9$ 中的系数.

9. 求 x^{18} 在 $(x+x^2+x^3+x^4+x^5+x^6+x^7)^4$ 中的系数.

10. 求 x^{36} 在 $(x^2+x^3+x^4+x^5+x^6+x^7+x^8)^6$ 中的系数.

11. 求 x^{12} 在下面的代数式中的系数:

(a) $x^2(1-x)^{-10}$; **(b)** $\dfrac{x^2-3x}{(1-x)^4}$; **(c)** $\dfrac{(1-x^2)^5}{(1-x)^5}$; **(d)** $\dfrac{x+3}{1-2x+x^2}$; **(e)** $\dfrac{b^m x^m}{(1-bx)^{m+1}}$.

12. 对于下面的代数式 (a) 和 (b), 给出类似于 (1) 的公式.

(a) $1+x^4+x^8+\cdots+x^{24}$. **(b)** $x^{20}+x^{40}+\cdots+x^{80}$.

13. 求 x^9 在 $(x^2+x^3+x^4+x^5)^5$ 中的系数.

14. 求 x^{18} 在 $(1+x^3+x^6+x^9+\cdots)^7$ 中的系数.

15. 求 x^{12} 在下面各式中的系数:

(a) $(1-x)^8$; **(b)** $(1+x)^{-1}$; **(c)** $(1+x)^{-8}$; **(d)** $(1-4x)^{-5}$; **(e)** $(1+x^3)^{-4}$.

16. 求 x^{25} 在 $(1+x^3+x^8)^{10}$ 中的系数.

17. 使用生成函数, 求从一堆红球、白球和蓝球中选出 10 个球的方法数, 如果:

(a) 其中每种颜色至少有两个球. **(b)** 其中至多有两个红球.

(c) 其中蓝色球的个数为偶数.

18. 使用生成函数, 求把 r 个果冻分给 8 个孩子的方法数, 如果:

(a) 每个孩子至少得到一个果冻. **(b)** 每个孩子得到偶数个果冻.

19. 假设有 5 种巧克力圣代, 那么有多少种方法预定 12 个巧克力圣代且每种冰激凌至多有 4 个?

20. 有多少种方法使用 5 种涂料粉刷一个旅馆的 10 间相同房间, 使得至多有 3 间房子涂成绿色, 至多有 3 个房间涂成蓝色, 至多有 3 个房间涂成红色, 而对其他两种颜色 (黑色和白色) 没有限制?

21. 有多少种方法把 20 美分分配给 n 个孩子和一位父母, 如果这个父母或者得到一个 5 美分硬币, 或者得到一个 10 美分硬币, 而且:

(a) 孩子们可以得到任意数量? **(b)** 每个孩子至多得到 1 美分?

22. 当投掷 10 枚不同的骰子时, 有多少种方法能够得到 25 的点数和?

23. 有多少种方法从 7 种巧力中选出 300 块, 如果每种类型都是 20 块一盒, 且每种巧克力至少选出一盒但不超过 5 盒? (提示: 用巧克力的盒数来求解这一问题.)

24. 如果我们认为来自同一个州的 2 名参议员是相同的 (共有 50 个州), 那么可以形成多少个有 40 名参议员的委员会?

25. 假设有 3 种图书, 第一种有 6 本, 第二种有 7 本, 第三种有 11 本, 有多少种方法把这些书分给两位老师, 使得每位老师得到 12 本且每种至少 2 本?

26. 有多少种方法把 5 个梨、5 个苹果、5 个面包圈、5 个棒棒糖、5 块巧克力和 5 块糖无序地分成两堆, 使得每堆有 15 件物品?

27. 有多少种方法从 4 个孩子和 6 个成人那里募集 24 美元, 使得每个人至少捐出 1 美元, 但是每个孩子至多捐出 4 美元, 而每个成人至多捐出 7 美元?

28. 如果把一枚硬币投掷 25 次, 且出现 8 次反面, 没有连续 6 个 (或更多) 正面出现的概率是多少?

29. 如果把 10 份牛排和 15 份龙虾分给 4 个人, 每个人分到的牛排最多 5 份, 最多分到 5 份

. 龙虾, 有多少种分法?

30. 假设投掷一枚骰子任意多次, 证明: 点数和等于 r 的方法数的生成函数是 $(1-x-x^2-x^3-x^4-x^5-x^6)^{-1}$.

31. 使用生成函数证明:

$$\sum_{k=0}^{r}\binom{m}{k}\binom{n}{r-k}=\binom{m+n}{r}$$

32. 使用方程

$$\frac{(1-x^2)^n}{(1-x)^n}=(1+x)^n$$

证明: 如果 $m\leqslant n$ 且 m 是偶数, 则

$$\sum_{k=0}^{m/2}(-1)^k\binom{n}{k}\binom{n+m-2k-1}{n-1}=\binom{n}{m}$$

33. 使用二项式展开计算:

(a) $\sum\limits_{k=0}^{m}\binom{m}{k}\binom{n}{r+k}$; **(b)** $\sum\limits_{k=0}^{r}(-1)^k\binom{n}{k}\binom{n}{r-k}$; **(c)** $\sum\limits_{k=0}^{n}2^k\binom{n}{k}$.

34. **(a)** 计算 $\sum\limits_{k=n_1}^{n_2}\binom{k+5}{k}$. **(b)** 计算 $\sum\limits_{k=0}^{m}\binom{n-k}{m-k}$.

35. **(a)** 证明 $\sum\limits_{k=0}^{\infty}\left(\frac{1}{2}\right)^k\binom{n+k-1}{k}=2^n$. **(b)** 计算 $\sum\limits_{k=0}^{\infty}\left(\frac{1}{2}\right)^k k$.

36. 为什么不能在公式 (5) 中令 $x=-1$ 来证明

$$\left(\frac{1}{2}\right)^n=\sum_{k=0}^{\infty}(-1)^k\binom{n+k-1}{k}$$

37. 设 $g(x)$ 是 a_k 的生成函数, 证明: $g^{(k)}(0)/k!=a_k$.

38. 离散随机变量 X 的*概率生成函数*(probability generating function)$P_X(t)$ 有多项式展开式, 其中 t^r 的系数 p_r 等于 $X=r$ 的概率.

(a) 设 X 是投掷一枚均匀硬币 n 次时正面出现的次数, 证明: $P_X(t)=(1/2)^n(1+t)^n$.

(b) 设 X 是投掷一枚偏差硬币 n 次时正面出现的次数, 且出现正面的概率是 p(且 $q=1-p$), 证明: $P_X(t)=(q+pt)^n$.

(c) 设 X 是投掷一枚均匀硬币直到第 5 个正面出现时的投掷次数, 求 $P_X(t)$.

(d) 设出现正面的概率是 p, 重做 (c), 直到第 m 次正面出现.

39. **(a)** 离散随机变量 X 的*期望值*(expected value)$E(X)$ 定义为 $\sum p_r r$. 证明: $E(X)=P_X'(1)$, 即 (d/dt)$P_X(t)$ 当 t 为 1 时的情况.

(b) 求练习 38 中的随机变量 X 的 $E(X)$.

40. **(a)** 离散随机变量 X 的*二阶矩*$E_2(X)$ 定义为 $\sum p_r r^2$. 证明 $E_2(X)=P_X'(1)+P_X''(1)$.

(b) 求练习 38(b) 和 38(c) 中的随机变量 X 的 $E_2(X)$.

41. 假设投掷一枚均匀硬币直到第 m 次正面出现, 并假设连续出现的反面个数不超过 s 个. 对于投掷次数 X, 求 $P_X(t)$.

42. 对于实验 A' 和 A'', 每次试验的成功概率分别是 p' 和 p'', 而且分别进行了 n' 次和 n'' 次试验. 设 X' 和 X'' 分别是在各自实验中成功的次数, 且设 X 是两个实验的总成功次数. 证明下面的等式:

(a) $P_X(t)=P_X'(t)P_X''(t)$; **(b)** $E(X)=E(X')+E(X'')$;

(c) $E_2(X)=E_2(X')+E_2(X'')+E(X'X'')$.

43. 假设我们投掷一枚红色的骰子一次, 然后再投掷一枚绿色的骰子, 投掷的次数等于红色骰子上的点数. a_r 是和等于 r 的绿色骰子投掷序列 (可变长度) 的方法数, 证明 a_r 的生成函数是 $f(f(x))$, 其中 $f(x) = (x + x^2 + x^3 + x^4 + x^5 + x^6)$.

44. 假设 X 是快餐服务前台接待一名顾客所需的分钟数的一个随机变量. 假设 Y 是一分钟内站在前台等待服务的人数的一个随机变量. 设 Z 是当一个人接受服务时站排的人数. 证明 $P_Z(t) = P_X(P_Y(t))$.

6.3 分 拆

本节讨论分拆及其生成函数. 遗憾的是, 没有简单的方法计算这其中大部分生成函数的系数.

一组 r 个相同对象的一个**分拆**(partition) 是把这一组对象分成各种大小的子集的一个组合. 类似地, 我们定义整数 r 的一个分拆是和等于 r 的正整数的一个组合. 形式上, 我们把这一组合写成和的形式, 且按递增顺序列出分拆中的整数. 例如, 整数 5 的 7 个分拆是

$$1+1+1+1+1 \qquad 1+1+1+2 \qquad 1+1+3 \qquad 1+2+2 \quad 1+4 \quad 2+3 \quad 5$$

注意, 5 自身是一个 “平凡” 分拆.

我们来构建 a_r 的生成函数, 其中 a_r 是整数 r 的分拆数. 整数的分拆可以描述成在和中有多少个 1, 有多少个 2, 等等. 设 e_k 表示分拆中 k 的个数. 那么有

$$1e_1 + 2e_2 + 3e_3 + \cdots + ke_k + \cdots + re_r = r$$

直观上, 我们可以想象从无数量限制的几堆对象中选出 r 个对象, 其中第一堆只包含一个对象, 第二堆包含粘成对的对象对, 第三堆包含粘成三元组的对象组, 以此类推. 为了用生成函数对这一方程整数解问题建模, 需要产生如下形式的形式积的多项式因子:

$$\left\{\begin{matrix} x^0 \\ x^1 \\ x^2 \\ \vdots \end{matrix}\right\}\left\{\begin{matrix} (x^2)^0 \\ (x^2)^1 \\ (x^2)^2 \\ \vdots \end{matrix}\right\}\left\{\begin{matrix} (x^3)^0 \\ (x^3)^1 \\ (x^3)^2 \\ \vdots \end{matrix}\right\}\cdots\left\{\begin{matrix} (x^k)^0 \\ (x^k)^1 \\ (x^k)^2 \\ \vdots \end{matrix}\right\}\cdots$$

生成函数 $g(x)$ 一定是

$$\begin{aligned} g(x) = &(1 + x + x^2 + x^3 + \cdots + x^n + \cdots) \\ &\cdot(1 + x^2 + x^4 + x^6 + \cdots + x^{2n} + \cdots) \\ &\cdot(1 + x^3 + x^6 + x^9 + \cdots + x^{3n} + \cdots) \\ &\qquad\vdots \\ &\cdot(1 + x^k + x^{2k} + x^{3k} + \cdots + x^{kn} + \cdots) \\ &\qquad\vdots \end{aligned}$$

如果设 $y=x^2$, 那么 $g(x)$ 中的第二个因子变成 $1+y+y^2+\cdots=(1-y)^{-1}$. 因此

$$1+x^2+x^4+x^6+\cdots+x^{2n}+\cdots=(1-x^2)^{-1}$$

类似地, 第 k 个因子可以写成 $(1-x^k)^{-1}$. 因为分拆到 $r=m$ 为止, 所以只需要前 m 个因子. 但是, r 可以任意取值, 我们需要无限多个多项式因子. 于是

$$g(x)=\frac{1}{(1-x)(1-x^2)(1-x^3)\cdots(1-x^k)\cdots}$$

现在考虑一些更特殊的分拆问题.

例 1 求 a_r 的生成函数, 其中 a_r 是把 r 表示成不同整数的和的方法数.

这里, 我们必须对一般分拆问题做限制, 使得其中任意整数都不得重复. 例如, 有 3 种方法把 5 写成不同整数的和：1+4, 2+3, 5. 对无限制分拆的生成函数作适当修改后得到：

$$g(x)=(1+x)(1+x^2)(1+x^3)(1+x^4)\cdots(1+x^k)\cdots \qquad \square$$

例 2 求 a_r 的生成函数, 其中 a_r 是选取 2 美分、3 美分和 5 美分使得和等于 r 美分的方法数.

这个问题等价于求下面方程的整数解的个数：

$$2e_2+3e_3+5e_5=r, \qquad 0\leqslant e_2,e_3,e_5$$

所求的生成函数是

$$(1+x^2+x^4+x^6+\cdots)(1+x^3+x^6+x^9+\cdots)\cdot(1+x^5+x^{10}+x^{15}+\cdots) \qquad \square$$

例 3 使用生成函数证明：每一个正整数都可以唯一地写成 2 的不同幂的和.

把一个整数 r 写成 2 的不同幂的和的方法数 a_r 的生成函数, 类似于例 1 中不同整数和的生成函数, 只不过现在使用的整数是 2 的幂. 这个生成函数是

$$g^*(x)=(1+x)(1+x^2)(1+x^4)(1+x^8)\cdots(1+x^{2^k})\cdots$$

为了证明每一个整数都可以唯一地写成 2 的不同幂的和, 必须证明在 $g^*(x)$ 中 x 的每个幂的系数都是 1. 即证明：

$$g^*(x)=1+x+x^2+x^3+\cdots=\frac{1}{1-x}$$

或等价地, 证明

$$(1-x)g^*(x)=1$$

通过反复使用因式分解 $(1-x^k)(1+x^k)=1-x^{2k}$ 来证明这一等式：

$$\begin{aligned}(1-x)g^*(x)&=(1-x)(1+x)(1+x^2)(1+x^4)(1+x^8)\cdots\\&=[\quad(1-x^2)\quad](1+x^2)(1+x^4)(1+x^8)\cdots\\&=[\qquad(1-x^4)\qquad]\quad(1+x^4)(1+x^8)\cdots\\&\qquad\vdots\\&=1\end{aligned}$$

通过持续使用替换 $(1-x^k)(1+x^k)=1-x^{2k}$, 最终我们消除 $(1-x)g^*(x)$ 中的所有因子. 形式上, $(1-x)g^*(x)$ 中任意特定的 x^k 的系数一定是 0. 所以 $(1-x)g^*(x)=1$, 即, $g^*(x)=1+x+x^2+\cdots$, 这正是我们需要的. □

Ferrers 示图是研究分拆的一种便利工具. Ferrers 示图以若干行显示数量逐渐减少的点来表示整数 r 的分拆. 图 6.1a 的 Ferrers 示图给出 15 的一个分拆 1+2+2+3+7. 如果把 r 的分拆的 Ferrers 示图中的行和列转置, 那么我们得到 r 的另一个分拆的 Ferrers 示图. 这个示图称为原来示图的**共轭**(conjugate). 例如, 图 6.1b 展示了图 6.1a 中的 Ferrers 示图的共轭. 这个新的 Ferrers 示图代表 15 的另一个分拆, 1+1+1+1+2+4+5. 显然, 转置是唯一的: 两个 Ferrers 示图有相同的共轭当且仅当它们是相同的.

(a)　(b)

图 6.1

例 4　证明: 把整数 r 分成 m 个正整数的和的分拆数, 等于把 r 分成最大数为 m 的正整数的和的分拆数.

如果画出把 r 分成 m 个部分的分拆的 Ferrers 示图, 那么这个 Ferrers 示图将有 m 行. 这样的示图的转置将有 m 列, 即最长的行将有 m 个点. 因此这两类分拆之间存在一一对应. □

练　习　6.3

练习概况　练习 1~10 是关于分拆的生成函数建模问题. 中间的 7 道练习使用 Ferrers 示图证明分拆恒等式. 其后的 5 道练习是较难的分拆问题.

1. 列出下面整数 6 的所有分拆: **(a)** 4;　**(b)** 6.

2. 求 a_r 的生成函数, 其中 a_r 是把 r 分成下面各种情况时的分拆数:

(a) 偶数个整数;　**(b)** 不同的奇数整数.

3. 求把整数 r 写成正整数的和, 且其中任意正整数的出现次数都不超过 3 次的方法数的生成函数.

4. 求方程 $2x+3y+7z=r$ 的整数解个数的生成函数, 其中:

(a) $x,y,z\geqslant 0$;　**(b)** $0\leqslant z\leqslant 2\leqslant y\leqslant 8\leqslant x$.

5. 求把 r 美分兑换成一美分硬币、五美分硬币、一角硬币和二角五分硬币时的方法数的生成函数.

6. 求把 r 个相同对象分配到下面各盒子里的方法数的生成函数:

(a) 3 个不可区分的盒子;　**(b)** n 个不可区分的盒子 $(n\leqslant r)$.

7. **(a)** 通过具体给出所有分拆, 证明: 把 10 分成不同部分 (整数) 的分拆数等于把 10 分成奇数个部分的分拆数.

(b) 用代数方法证明: 把 r 分成不同部分的分拆的生成函数, 等于把 r 分成奇数个部分的分拆的生成函数, 因此这两种类型的分拆数相等.

8. 使用生成函数证明：每一个正整数都有唯一的十进制表示.

9. **(a)** 使用归纳法证明例 3 中的结果.

(b) 通过在 $g^*(x)$ 中递归地替换 $(1+x^k)=(1-x^{2k})/\ (1-x^k)$, 直接证明例 3 中的结果.

10. 例 3 中的方程 $g^*(x)(1-x)=1$ 可以重写成 $1-x=1/\ g^*(x)$. 使用后面的方程证明：整数 r 的所有 2 的幂的分拆中 $(r\geqslant 2)$, 有奇数个部分的分拆数等于有偶数个部分的分拆数.

11. 设 $R(r,k)$ 表示把整数 r 分成 k 部分的分拆数.

(a) 证明：$R(r,k)=R(r-1,k-1)+R(r-k,k)$.

(b) 证明：$\sum\limits_{k=1}^{r}R(n-r,k)=R(n,r)$.

12. 使用 Ferrers 示图证明：把一个整数分成偶数大小的部分的分拆数, 等于每个部分出现的次数为偶数的分拆数.

13. 把整数乘法 mn, “m 乘以 n”, 解释为 m 个 n 的和. 证明：$mn=nm$.

14. 证明：把整数 n 分成 3 个部分的分拆数, 等于把 $2n$ 分成各部分大小 $<\ n$ 的 3 个部分的分拆数.

15. 证明：n 的分拆数等于把 $2n$ 分成 n 个部分的分拆数.

16. 证明：把 $2r+k$ 分成 $r+k$ 个部分的分拆数等于 k.

17. 证明：把 $r+k$ 分成 k 个部分的分拆数等于：

(a) 把 $r+\binom{k+1}{2}$ 分成 k 个不同部分的分拆数; **(b)** 把 r 分成大小 $\leqslant k$ 的部分的分拆数.

18. 证明：n 的有序分拆数是 2^{n-1}. 例如, 4 的有序分拆如下所示：4, 1+3, 2+2, 3+1, 1+1+2, 1+2+1, 2+1+1, 1+1+1+1(提示：把 n 个 1 写成一行, 并确定把这个序列分成若干簇 1 的所有方法数.)

19. **(a)** 求 a_n 的生成函数, 其中 a_n 是和至多等于 n 的分拆数.

(b) 求 a_n 的生成函数, 其中 a_n 是把 n 分成 3 个部分, 其中任意部分都小于等于其他两部分之和的分拆数.

(c) 求 a_n 的生成函数, 其中 a_n 是有整数边长且周长为 n 的不同 (不全等) 三角形的数量.

20. 证明：$2(1-x)^{-3}[(1-x)^{-3}+(1+x)^{-3}]$ 是投掷 r 枚相同的骰子并得到偶数和的方法数的生成函数.

21. **(a)** 整数 r 的分拆称为是自共轭的 (self-conjugate), 如果这个分拆的 Ferrers 示图与其自身的转置相同. 求 r 的自共轭分拆与把 r 分成不同奇数部分的分拆之间的一一对应.

(b) Ferrers 示图的左上角的点构成的最大方形称为这个 Ferrers 示图的 Durfee*方形*. 求 r 的 Durfee 方形的大小为 $k(k^*k$ 的点矩阵) 的自共轭分拆数的生成函数. (提示：使用这些分拆与把 $r-k^2$ 分成偶数个、大小最多为 $2k$ 的部分的分拆之间的一一对应关系.)

(c) 证明：$(1+x)(1+x^3)(1+x^5)\cdots=1+\sum\limits_{k=1}^{\infty}\dfrac{x^{k^2}}{(1-x^2)(1-x^4)(1-x^6)\cdots(1-x^{2k})}$

22. 设 $\left\{{n\atop k}\right\}$表示把 n 个不同对象分成 k 个非空子集的分拆数. 证明：

$$\left\{{n+1\atop k}\right\}=k\left\{{n\atop k}\right\}+\left\{{n\atop k-1}\right\}.$$

23. 编写一个计算机程序, 确定把整数 r 分成下面要求的部分的所有分拆的数量：

(a) 分成 k 部分; **(b)** 分成任意多部分; **(c)** 分成不同部分.

6.4 指数生成函数

本节讨论指数生成函数. 我们使用指数生成函数对重复排列问题建模并求解. 前几节所使用的对重复选择问题进行建模的生成函数称为**普通生成函数**(ordinary generating function). 指数生成函数包含更复杂的建模过程, 比普通生成函数的建模过程要复杂得多. 考虑这样的问题: 从无穷多个 a, b, c 中选出个 4 个字母组成的至少包含两个 a 的不同的单词 (不同的排列), 这些单词的数量有多少. 形成这样的单词的可能 4 字母选择是{a, a, a, a}, {a, a, a, b}, {a, a, a, c}, {a, a, b, b}, {a, a, b, c}, 以及{a, a, c, c}. 这 6 个选择所形成的可能排列数分别是

$$\frac{4!}{4!0!0!} \quad \frac{4!}{3!1!0!} \quad \frac{4!}{3!0!1!} \quad \frac{4!}{2!2!0!} \quad \frac{4!}{2!1!1!} \quad \frac{4!}{2!0!2!}$$

所以, 单词的总数是上面 6 项之和.

这样的单词所使用的字母选取包含从若干个 a、若干个 b 和若干个 c 中重复选取至少有两个 a 的所有 4 字母子集. 即, 这样的选取的数量等于下面方程的整数解个数:

$$e_1 + e_2 + e_3 = 4, \quad \text{其中} 2 \leqslant e_1, \quad \text{且} 0 \leqslant e_2, e_3 \tag{1}$$

初看起来, 4 字母单词的问题好像类似于前面可以用 (普通) 生成函数建模的问题. 然而, 对于本节的情况, 我们不是把 (1) 的每个整数解在可能单词的计数中计为 1. 实际上, 每个整数解必须贡献 $4!/(e_1!e_2!e_3!)$ 个单词. 用生成函数表示, x^4 的系数将计数带有下面系数的所有形式积:

$$\frac{(e_1+e_2+e_3)!}{e_1!e_2!e_3!} x^{e_1} x^{e_2} x^{e_3} \quad 2 \leqslant e_1, \quad 0 \leqslant e_2, e_3$$

其中, 上式中 x 的指数和等于 4(一个更困难的问题). 幸运的是, 指数生成函数正好产生这种形式的形式积.

a_r 的**指数生成函数**(exponential generating function)$g(x)$ 是有如下幂级数展开式的函数, 其中 a_r 是 r 个对象的排列数.

$$g(x) = a_0 + a_1 x + a_2 \frac{x^2}{2!} + a_3 \frac{x^3}{3!} + \cdots + a_r \frac{x^r}{r!} + \cdots$$

我们用构建普通生成函数的方法构建指数生成函数: 每种对象有一个多项式因子; 每个因子有一组 x 的幂, 这些幂是这种对象的选取个数的目录. 然而, 此时每个幂 x^r 要除以 $r!$.

作为一个例子, 考虑至少有两个 a 的 4 字母单词问题. 我们说, 无穷多个 a, b, c 所形成的至少包含两个 a 的 r 字母单词的个数的指数生成函数是:

$$g(x) = \left(\frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \cdots\right)\left(1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \cdots\right)^2 \tag{2}$$

x^r 在 (2) 中的系数是所有积 $(x^{e_1}/e_1!)(x^{e_2}/e_2!)(x^{e_3}/e_3!)$ 的和, 其中 $e_1+e_2+e_3=r, 2\leqslant e_1, 0\leqslant e_2, e_3$. 如果用 $r!$ 去除 x^r, 然后再乘以 $r!$ 来保持原式成立, 于是 $g(x)$ 中的 x^r 项变成

$$\left(\sum_{e_1+e_2+e_3=r}\frac{r!}{e_1!e_2!e_3!}\right)\frac{x^r}{r!},\qquad 2\leqslant e_1,\quad 0\leqslant e_2,e_3$$

$x^r/r!$ 的系数正是我们想要的. 所以, 这样的 r 字母单词的指数生成函数的确是展开式 (2). 使指数生成函数发挥作用的是把 x^r 除以 $r!$, 然后再把 x^r 的系数乘以 $r!$ 的"技巧".

在说明如何计算指数生成函数的系数之前, 先给出另外几个指数生成函数建模的例子.

例 1　求 a_r 的指数生成函数, 其中 a_r 是 n 个对象的无重复 r 排列数.

我们知道这一问题的答案是 $P(n,r)$. 因为没有重复, 所以指数函数是 $(1+x)^n$, 这就是二项式. x^r 在 $(1+x)^n$ 中的系数是 $\binom{n}{r}$. 然而, 现在想要的是 $x^r/r!$ 在 $(1+x)^n$ 中的系数:

$$\binom{n}{r}x^r=\frac{n!}{(n-r)!r!}x^r=\frac{n!}{(n-r)!}\frac{x^r}{r!}$$

因此, $a_r=n!/(n-r)!=P(n,r)$, 这正是我们要求的. □

例 2　求 a_r 的指数生成函数, 其中 a_r 是从 4 个不同类型的对象中选出 r 个对象且每种对象至少出现 2 次但不超过 5 次的不同排列数.

排列 r 个对象且第 i 种类型的对象有 e_i 个的方法数是 $r!/e_1!e_2!e_3!e_4!$. 为了求满足条件限制的所有项之和, 我们要求 $x^r/r!$ 在 $(x^2/2!+x^3/3!+x^4/4!+x^5/5!)^4$ 中的系数. □

例 3　求把 r 个人安置在 3 个不同房间且每个房间至少有 1 个人的方法数的指数生成函数. 当每个房间的人数都是偶数时, 重解这一问题.

回想一下, 不同对象的分配等价于重复排列; 参考 5.4 节. 所求的指数生成函数是:

$$\left(x+\frac{x^2}{2!}+\frac{x^3}{3!}+\frac{x^4}{4!}+\cdots\right)^3\quad 和\quad \left(1+\frac{x^2}{2!}+\frac{x^4}{4!}+\cdots\right)^3$$ □

与指数生成函数相关联的等式或者展开式很少. 因此, 只有有限个容易估算系数的指数生成函数. 指数生成函数的基本展开式是:

$$\mathrm{e}^x=1+x+\frac{x^2}{2!}+\frac{x^3}{3!}+\cdots+\frac{x^r}{r!}+\cdots \tag{3}$$

在 (3) 中用 nx 取代 x, 我们得到:

$$\mathrm{e}^{nx}=1+nx+\frac{n^2x^2}{2!}+\frac{n^3x^3}{3!}+\cdots+\frac{n^rx^r}{r!}+\cdots \tag{4}$$

(3) 中的幂级数是 e^x 的泰勒级数, 这一幂级数在所有微积分课本中都可以看到. 这个展开式和 e^{nx} 的伴随展开式对所有 x 值都成立. 无法提取 x 在指数生成函数中

的公共幂, 因为 x 的幂必须与分母中的 $r!$ 配对. 因此, 所能做的事情就是从 e^x 中减去 x 的不存在的 (最低) 幂. 例如, 表示两个或更多特定类型对象的指数因子可以写成:

$$\frac{x^2}{2!}+\frac{x^3}{3!}+\frac{x^4}{4!}+\cdots=\mathrm{e}^x-1-x$$

从 (3) 可以得到的两个有用的展开式是:

$$\frac{1}{2}(\mathrm{e}^x+\mathrm{e}^{-x})=1+\frac{x^2}{2!}+\frac{x^4}{4!}+\frac{x^6}{6!}+\cdots \tag{5}$$

$$\frac{1}{2}(\mathrm{e}^x-\mathrm{e}^{-x})=x+\frac{x^3}{3!}+\frac{x^5}{5!}+\frac{x^7}{7!}+\cdots \tag{6}$$

我们使用展开式 (3) 到 (6) 解决几个例子.

例 4 求从 n 种类型的无限多个对象中选出 r 个对象的不同排列数.

在第 5 章, 我们是这样解决这个问题的: 在 r 个位置中的每一个位置上有 n 种类型的对象可以选择, 所以总共有 n^r 种排列. 现在用指数生成函数解决这个问题. 这个问题的指数生成函数是:

$$\left(1+x+\frac{x^2}{2!}+\frac{x^3}{3!}+\cdots\right)^n=(\mathrm{e}^x)^n=\mathrm{e}^{nx}$$

根据 (4), 在这个生成函数中 $x^r/r!$ 的系数是 n^r. □

例 5 求把 25 个人安置到 3 个房间且每个房间至少有 1 个人的方法数.

在例 3 中, 我们发现这个问题的指数生成函数是:

$$\left(x+\frac{x^2}{2!}+\frac{x^3}{3!}+\cdots\right)^3=(\mathrm{e}^x-1)^3$$

为了求 $x^r/r!$ 在 $(\mathrm{e}^x-1)^3$ 中的系数, 首先必须用 e^x 展开这个二项式表达式:

$$(\mathrm{e}^x-1)^3=\mathrm{e}^{3x}-3\mathrm{e}^{2x}+3\mathrm{e}^x-1$$

由 (4), 我们得到:

$$\mathrm{e}^{3x}-3\mathrm{e}^{2x}+3\mathrm{e}^x-1=\sum_{r=0}^{\infty}3^r\frac{x^r}{r!}-3\sum_{r=0}^{\infty}2^r\frac{x^r}{r!}+3\sum_{r=0}^{\infty}\frac{x^r}{r!}-1$$

所以 $x^{25}/25!$ 的系数是 $3^{25}-(3\times 2^{25})+3$. □

例 6 求有偶数个 0 和奇数个 1 的 r 位四进制序列 (其数字是 0, 1, 2, 3) 的个数.

这一问题的指数生成函数是:

$$\left(1+\frac{x^2}{2!}+\frac{x^4}{4!}+\frac{x^6}{6!}+\cdots\right)\left(x+\frac{x^3}{3!}+\frac{x^5}{5!}+\frac{x^7}{7!}+\cdots\right)\left(1+x+\frac{x^2}{2!}+\frac{x^3}{3!}+\cdots\right)^2$$

使用等式 (5) 和 (6), 可以把这个表达式写成:

$$\frac{1}{2}(\mathrm{e}^x+\mathrm{e}^{-x})\times\frac{1}{2}(\mathrm{e}^x-\mathrm{e}^{-x})\mathrm{e}^x\mathrm{e}^x=\frac{1}{4}(\mathrm{e}^{2x}-\mathrm{e}^0+\mathrm{e}^0-\mathrm{e}^{-2x})\mathrm{e}^x\mathrm{e}^x$$

$$= \frac{1}{4}(e^{2x} - e^{-2x})e^{2x} = \frac{1}{4}(e^{4x} - 1) = \frac{1}{4}\left(\sum_{r=0}^{\infty} 4^r \frac{x^r}{r!} - 1\right)$$

于是, 对于 $r > 0$, $x^r/r!$ 的系数是 $\frac{1}{4} \times 4^r = 4^{r-1}$. 这个问题的答案的简单形式表明, 应该存在某种组合讨论来直接得到这样简洁的答案. □

练 习 6.4

练习概况 前 15 道练习大部分都与本节的例子类似. 练习 21 和练习 22 是 6.2 节中的概率生成函数的延伸.

1. 求从 5 种对象中选出 r 个对象且每种类型至多为 5 个的排列数的指数生成函数.

2. 把 r 个人分配到 6 个不同的房间且每个房间的人数是 2 个人到 4 个人, 求这种分配的方法数的指数生成函数.

3. 求 a_r 的指数生成函数, 其中 a_r 是每个元音至多使用一次 (辅音可以重复) 的 r 字母单词的个数的指数生成函数.

4. **(a)** 求 $s_{n,r}$ 的指数生成函数. 其中 $s_{n,r}$ 是把 r 个不同的对象分配到 n 个不同的盒子且没有空盒子的方法数. n 是一个固定的常量.

(b) 确定 $s_{n,r}$. 数 $s_{n,r}/n!$ 称为第二类 Stirling 数.

5. 对于从 52 张牌中发 13 张牌的序列的方法数, 并设不计花色, 只计牌值, 求其指数生成函数并确定相应的系数.

6. 如果第一个孩子至少得到 2 个玩具, 那么有多少种方法把 8 个不同玩具分配给 4 个孩子?

7. 有多少个 r 位三进制序列, 使得:

(a) 有偶数个 0? **(b)** 有偶数个 0 且有偶数个 1?

(c) 至少有一个 0 和至少一个 1?

8. 使用 1 美分、5 美分、10 美分和 25 美分硬币可以形成多少个至少有一个 1 美分硬币和奇数个 25 美分硬币的排列? (名字相同的硬币是相同的.)

9. 有多少个 10 字母单词, 使得字母 e, n, r, s 中的每一个字母:

(a) 至多出现一次? **(b)** 至少出现一次?

10. 有多少 r 位三进制序列, 使得:

(a) 没有正好出现两次的数字? **(b)** 0 和 1 都出现正偶数次?

11. 0 和 1 的总数是偶数的 r 位四进制序列有多少个?

12. 把 r 个不同对象分配到 5 个不同盒子且 $b_1 < b_2 \leqslant 4$, 其中 b_1 和 b_2 分别是盒子 1 和盒子 2 中的对象个数. 求这一分配方法数的指数生成函数.

13. **(a)** 当 r 个不同对象随机分配到 n 个不同的盒子里时, 前两个盒子都至少有一个对象的概率为 p_r. 求 p_r 的指数生成函数.

(b) 确定 p_r.

14. 求把 r 个不同对象分配到 n 个不同的盒子且正好有 m 个非空盒子的分配数的指数生成函数.

15. 求下列各 a_r 的指数生成函数:

(a) $a_r = 1/(r+1)$; **(b)** $a_r = r!$.

16. 证明：如果 $g(x)$ 是 a_r 的指数生成函数，那么 $g^{(k)}(0)=a_k$，其中 $g^{(k)}(x)$ 是 $g(x)$ 的第 k 阶导数.

17. 如果有

$$f(x)=\sum_{r=0}^{\infty}a_r\frac{x^r}{r!},\qquad g(x)=\sum_{r=0}^{\infty}b_r\frac{x^r}{r!}$$

且有

$$h(x)=f(x)g(x)=\sum_{r=0}^{\infty}c_r\frac{x^r}{r!}$$

那么证明

$$c_r=\sum_{k=0}^{\infty}\binom{r}{k}a_kb_{r-k}$$

18. 通过对 e^x 的展开式和 e^y 的展开式做形式积来证明 $\mathrm{e}^x\mathrm{e}^y=\mathrm{e}^{x+y}$.

19. 寻求一个组合论证来说明为什么练习 6 中的答案是 4^{r-1}.

20. 选出 r 个不同对象的某个子集 (可以是空集)，然后再把它们分配到 n 个不同的盒子里，且考虑在每个盒子中的顺序. 证明这一分配的方法数的指数生成函数是 $\mathrm{e}^x/(1-x)^n$.

21. 泊松随机变量 X 有 $p_r=\mathrm{Prob}(x=r)=\frac{\mu^r}{r!}\mathrm{e}^{-\mu}$. 求 X 的概率生成函数 (参见 6.2 节练习 34).

22. 设 $P(x)=\sum\limits_{k=0}^{\infty}p_kx^k$ 是离散随机变量 X 的概率生成函数，即，p_k 是 $X=k$ 的概率.

(a) 证明：X 的第 k 个矩 $m_k=\sum\limits_{j=0}^{\infty}j^kp_j$ 的指数生成函数是 $P(\mathrm{e}^x)$.

(b) 我们定义 X 的第 k 个阶乘矩 m_k^* 等于 $\sum\limits_{j=k}^{\infty}j!/(j-k)!p_j$. 证明：$m_k^*$ 的指数生成函数是 $P(x+1)$.

(c) 设 X 是投掷 n 枚硬币时正面朝上的个数，求 m_1, m_2, m_2^*. (提示：使用练习 16.)

(d) 如果 X 是泊松随机变量 (参见练习 21)，求 m_1 和 m_1^*.

6.5 一个求和方法

本节说明如何构建普通生成函数 $h(x)$，它的 x^r 的系数是 r 的某个特殊函数 $p(r)$，诸如 r^2 或者 $C(r,3)$. 然后，对于每一个正数 n，我们使用 $h(x)$ 计算和 $p(0)+p(1)+\cdots+p(n)$. 下面 4 个简单规则是从已有的生成函数构建新生成函数的规则，我们将在本节反复使用它们. 假设 $A(x)=\sum a_nx^n$, $B(x)=\sum b_nx^n$, $C(x)=\sum c_nx^n$.

规则1. 如果 $b_n=da_n$，那么，对于任意常数 d, $B(x)=dA(x)$.

规则2. 如果 $c_n=a_n+b_n$，那么，$C(x)=A(x)+B(x)$.

规则3. 如果 $c_n=\sum\limits_{i=0}^{n}a_ib_{n-i}$，那么，$C(x)=A(x)B(x)$.

规则4. 如果除对 $i<k, b_i=0$ 之外，$b_n=a_{n-k}$，那么，$B(x)=x^kA(x)$.

规则 (3) 就是 6.2 节的表达式 (6). 其他规则都是显然的.

构建系数的其他基本操作是用 r 去乘生成函数 $g(x)$ 中的每一个系数. 通过对 $g(x)$ 求导数，然后乘以 x，就可以得到 $g^*(x)$，其中 $a_r^*=ra_r$，即，$g^*(x)=x\left(\frac{\mathrm{d}}{\mathrm{d}x}g(x)\right)$. 如果有

$$g(x) = a_0 + a_1x + a_2x^2 + a_3x^3 + \cdots + a_rx^r + \cdots \tag{1}$$

那么, $g(x)$ 的导数是

$$\frac{\mathrm{d}}{\mathrm{d}x}g(x) = a_1 + 2a_2x + 3a_3x^2 + \cdots + ra_rx^{r-1} + \cdots \tag{2}$$

把上式乘以 x(规则 4) 得到

$$g^*(x) = x\left[\frac{\mathrm{d}}{\mathrm{d}x}g(x)\right] = a_1x + 2a_2x^2 + 3a_3x^3 + \cdots + ra_rx^r + \cdots \tag{3}$$

注意, $g(x)$ 中的 a_0 项在 (2) 中消失, 因为 $0a_0 = 0$.

把这一操作与规则 (1) 和 (2) 结合起来, 我们可以反复地用 r 或一个常量去乘 x^r 的系数, 再把这样的数加起来, 从而得到一个 r 的多项式.

现在, 一个自然的问题是, 我们对什么样的系数使用这些操作? 毫无疑问, 这一答案当然是, 最开始是使下面生成函数的系数 a_r=1:

$$\frac{1}{1-x} = 1 + x + x^2 + x^3 + \cdots + x^r + \cdots$$

例 1 构建 $a_r = 2r^2$ 的生成函数 $h(x)$.

从 $1/(1-x)$ 开始, 我们使用等式 (2) 和 (3) 给出的生成函数操作, 首先把它的系数乘以 r, 得到:

$$x\left(\frac{\mathrm{d}}{\mathrm{d}x}\frac{1}{1-x}\right) = x\left(\frac{1}{(1-x)^2}\right) = 1x + 2x^2 + 3x^3 + \cdots + rx^r + \cdots$$

现在, 对 $x/(1-x)^2$ 重复使用这些操作, 得到:

$$x\left(\frac{\mathrm{d}}{\mathrm{d}x}\frac{x}{(1-x)^2}\right) = \frac{x(1+x)}{(1-x)^3} = 1^2x + 2^2x^2 + 3^2x^3 + \cdots + r^2x^r + \cdots$$

最后, 两边乘以 2, 得到我们需要的生成函数:

$$h(x) = \frac{2x(1+x)}{(1-x)^3} = (2\times 1^2)x + (2\times 2^2)x^2 + (2\times 3^2)x^3 + \cdots + (2\times r^2)x^r + \cdots$$

□

例 2 构建 $a_r = (r+1)r(r-1)$ 的生成函数 $h(x)$.

把 $(r+1)r(r-1)$ 乘开, 我们得到 $a_r = r^3 - r$, 如例 1 那样, 得到 r^3 和 r 的生成函数. 然后, 从一个生成函数减去另一个生成函数. 然而, 如果从 $3!(1-x)^{-4}$ 开始会更容易些, 它的系数 a_r 等于:

$$a_r = 3!\binom{r+4-1}{r} = 3!\frac{(r+3)!}{r!3!} = \frac{(r+3)!}{r!} = (r+3)(r+2)(r+1)$$

然后, $3!(1-x)^{-4}$ 的幂级数展开式是:

$$\frac{3!}{(1-x)^4} = (3\times 2\times 1) + (4\times 3\times 2)x + (5\times 4\times 3)x^2 + \cdots + (r+3)(r+2)(r+1)x^r + \cdots \qquad (4)$$

把 (4) 与要求的生成函数 $h(x)$ 比较:

$$h(x) = (3\times 2\times 1)x^2 + (4\times 3\times 2)x^3 + (5\times 4\times 3)x^4 + \cdots + (r+1)r(r-1)x^r + \cdots$$

该函数就是 (4) 中的级数乘以 x^2. 所以, $h(x) = 3!x^2(1-x)^{-4}$. □

扩展例 2 中的构建方法, 我们看到 $(n-1)!(1-x)^{-n}$ 有下面系数:

$$a_r = (n-1)!C(r+n-1,r) = [r+(n-1)][r+(n-2)]\cdots(r+1)$$

任意的递减项积的系数, 诸如 $(r+1)r(r-1)$, 都可以像例 2 那样由 $(n-1)!(1-x)^{-n}$ 构建出来, 其中 $n-1$ 是乘积中的项数.

迄今为止, 有指定系数的 $h(x)$ 的构建只是生成函数的代数操作的练习. 下面这个很容易证明的定理指出这个练习的目的.

定理 如果 $h(x)$ 是一个生成函数, 其中 a_r 是 x^r 的系数, 那么, $h^*(x) = h(x)/(1-x)$ 是 a_r 的和的生成函数. 即,

$$h^*(x) = a_0 + (a_0+a_1)x + (a_0+a_1+a_2)x^2 + \cdots + \left(\sum_{i=0}^{r} a_i\right)x^r + \cdots$$

从规则 (3) 出发, 可以得到积 $h^*(x) = f(x)h(x)$ 的系数, 其中 $f(x) = 1/(1-x)$, 从而证明这个定理. 现在, 重新回到前面的例子.

例 1(续) 计算和 $2\times 1^2 + 2\times 2^2 + 2\times 3^2 + \cdots + 2n^2$.

在例 1 中, 我们已经知道 $a_r = 2r^2$ 的生成函数是 $h(x) = 2x(1+x)/(1-x)^3$. 然后, 根据上面定理, 要求的和 $a_1 + a_2 + \cdots + a_n$ 是 x^n 在下面的生成函数中 x^n 的系数:

$$h^*(x) = h(x)/(1-x) = 2x(1+x)/(1-x)^4 = 2x(1-x)^{-4} + 2x^2(1-x)^{-4}$$

x^n 在 $2x(1-x)^{-4}$ 中的系数是 x^{n-1} 在 $2(1-x)^{-4}$ 中的系数, 而且 x^n 在 $2x^2(1-x)^{-4}$ 中的系数是 x^{n-2} 在 $2(1-x)^{-4}$ 中的系数. 因此, 给定的和等于

$$2\binom{(n-1)+4-1}{(n-1)} + 2\binom{(n-2)+4-1}{(n-2)} = 2\binom{n+2}{3} + \binom{n+1}{3} \qquad □$$

例 2(续) 计算和 $3\times 2\times 1 + 4\times 3\times 2 + \cdots + (n+1)n(n-1)$.

在例 2 中, 我们已经知道 $a_r = (r+1)r(r-1)$ 的生成函数 $h(x) = 6x^2(1-x)^{-4}$. 根据上面定理, 要求的和是 x^n 在 $h^*(x) = h(x)/(1-x) = 6x^2(1-x)^{-5}$ 中的系数. 这个系数是 $6(1-x)^{-5}$ 中的 x^{n-2} 项, 即, $6C((n-2)+5-1, n-2) = 6C(n+2,4)$.□

注意, 可以利用 5.5 节基于二项式等式的方法来解决这两个求和问题. 为了便于比较, 参见 5.5 节例 4 中所给出的对于这个和的分析 [注意 5.5 节的这个和的一般项是 $(n-2)(n-1)n$, 而不是 $(n+1)n(n-1)$]. 这两种方法各有长处.

练 习 6.5

1. 求下面各系数等于 a_r 的普通生成函数:
   (a) r; (b) 13; (c) $3r^2$; (d) $3r+7$; (e) $r(r-1)(r-2)(r-3)$.
2. (使用生成函数) 计算下面的和:
   (a) $0+1+2+\cdots+n$; (b) $13+13+\cdots+13$; (c) $0+3+12+\cdots+3n^2$;
   (d) $7+10+13+\cdots+(3n+7)$; (e) $4\times3\times2\times1+5\times4\times3\times2+\cdots+n(n-1)(n-2)(n-3)$.
3. 求 $a_r=r(r+2)$ 的生成函数 (不要把 r^2 和 $2r$ 的生成函数加在一起).
4. (a) 说明如何把 r^2 和 r^3 写成 $P(r,3)$、$P(r,2)$ 和 $P(r,1)$ 的线性组合.
   (b) 使用问题 (a) 求 $3r^3-5r^2+4r$ 的生成函数.
5. 求下面各 a_r 的生成函数:
   (a) $a_r=(r-1)^2$; (b) $a_r=1/r$.
6. 如果 $h(x)$ 是 a_r 的普通生成函数, x^r 在 $h(x)(1-x)$ 的系数是什么? (用 a_r 给出你的答案.)
7. 证明本节的定理 1.
8. 如果 $h(x)$ 是 a_r 的普通生成函数, 求 $s_r=\sum\limits_{k=r+1}^{\infty}a_k$ 的生成函数, 假设所有的 s_r 都是有限的, 且当 $r\to\infty$ 时, $a_r\to 0$.

6.6 小结及参考文献

生成函数曾经是计数问题既单纯又精巧的数学模型: 说它单纯是因为多项式乘法是为人熟知的, 这部分内容是高中代数中比较好理解的部分; 而说它精巧是因为在对生成函数做普通的代数操作后, 我们可以解决复杂的计数问题. 这些代数处理自动地为我们执行正确的组合推理. 注意, 生成函数是当代数学研究前沿中最普遍存在的代数方法的基本例子. 无论是为组合信息建模, 还是为几何信息或函数信息建模, 代数表达式都为我们完成现代数学的大部分工作.

本章使用生成函数对带有限制的重复组合和排列建模. 我们还为分拆问题建模 (但是没有解决). 最后, 我们展示了如何构建 x^r 的系数为 r 的给定函数的生成函数, 并使用这些生成函数计算相关的和. 在接下来的 3 章, 我们将使用生成函数模拟并解决其他组合问题. 6.2 节的练习 38 介绍了生成函数在概率理论中的运用 (关于更详细的内容, 参见 Feller[2]). 分析学中的拉普拉斯变换和傅立叶变换也是生成函数 [我们可以把一个函数 $f(t)$ 的傅立叶变换看成是 $f(t)$ 的傅立叶系数的生成函数].

De Moivre 于 1720 年前后首先使用了组合生成函数. 他使用生成函数得到了斐波纳契数的一个公式 (7.5 节给出这一公式的推导过程). 1748 年, 欧拉在他的著作中对分拆问题使用了生成函数. 而他同时对概率生成函数的工作是 18 世纪后期发展起来的组合生成函数理论的原始推动力. (参见 6.2 节的练习 38~42, 及 6.4 节

的练习 21 和练习 22.) 拉普拉斯对这两个理论都做出了很多贡献, 而且在他在 1812 年的经典著作《概率分析》中首先对二者给出了充分的探讨.

关于生成函数在组合数学中的应用的详细讨论, 参见 MacMahon[3]. 关于分拆问题的更好的介绍, 还可以参见 Cameron[1].

1. P. Cameron, *Combinatorics: Topics, Techniques, Algorithms*, Cambridge University Press, Cambridge, 1994.①

2. W. Feller, *An Introduction to Probability Theory and Its Applications*, vol. I, 2nd ed., John Wiley & Sons, New York, 1957.②

3. P. MacMahon, *Combinatory Analysis*, vols. I and II (1915), reprinted in one volume, Chelsea Publishing, New York, 1960.

① 人民邮电出版社即将出版本书的英文影印版. —— 编者注

② 人民邮电出版社于 2006 年出版了本书的中文版. —— 编者注

第 7 章　递 推 关 系

7.1　递推关系模型

本章说明如何用递推关系对各种计数问题建模. 然后, 我们讨论解决几种通用的递推关系的方法.

递推关系(recurrence relation)　是这样的递推公式, 对于涉及 n 个对象的过程, 它通过涉及更少对象的同一过程的方法数来计数这一过程的方法数. 即, 如果 a_k 是对 k 个对象做这一过程的方法数, 其中 $k=0,1,2,\cdots$, 那么递推关系是一个等式, 它把 a_n 表示成前若干个 a_k 的某个函数, $k<n$. 最简单的递推关系是诸如式 $a_n=2a_{n-1}$ 这样的等式. 下面的等式给出了若干递推关系形式, 我们使用它们对本章中的计数问题建模.

$$
\begin{aligned}
a_n &= c_1a_{n-1}+c_2a_{n-2}+\cdots+c_ra_{n-r}, \quad \text{其中 } c_i \text{ 是常数}\\
a_n &= ca_{n-1}+f(n), \quad \text{其中 } f(n) \text{ 是 } n \text{ 的某个函数}\\
a_n &= a_0a_{n-1}+a_1a_{n-2}+\cdots+a_{n-1}a_0\\
a_{n,m} &= a_{n-1,m}+a_{n-1,m-1}
\end{aligned}
$$

数学归纳法是证明公式或论断的证明技巧, 它对递增的 n 值归纳地检查公式或论断的正确性, 所以递推关系是解决枚举问题的一种计数技术, 它相继地对更大的 n 值递推地计算出答案.

细心的读者应该还记得, 数学归纳法包含一个初始步骤, 对 n 的某个 (最小的) 开始值验证公式或论断. 这种情况对递推公式也是一样. 只有在给定了某些初始值的前提下, 才可能递推地计算下一个 a_n. 如果递推关系的右侧包含前面 r 个 a_k, 那么就需要知道前 r 个值, $a_0,a_1,\cdots,a_{r-1}$. 例如, 对于关系 $a_n=a_{n-1}+a_{n-2}$, 仅仅知道 a_0=2 是不够的. 如果还给定 a_1=3, 那么就可以使用这个关系得到 $a_2=a_1+a_0=3+2=5, a_3=a_2+a_1=5+3=8, a_4=8+5=13$, 以此类推. 计算递推关系所需要的这些初始信息称为**初始条件**(initial condition).

如果能够设计一个递推关系来模拟我们正在研究的计数问题, 而且还确定了初始条件, 那么对于适当大小的 n, 例如 $n=20$, 我们可以通过递推地计算相继的 a_n 值, 直到想要的 n, 来很快地解决这一问题. 对于更大的 n 值, 则需要可编程的计算器或计算机.

对于很多通用类型的递推关系, a_n 有明确的公式. 7.2 节到 7.4 节讨论了这些递推关系的一些解决方案. 然而, 在通常情况下, 递推地确定 $a_0,a_1\cdots$, 直到想要的

值, 比如 a_{12}, 要比从 a_n 的复杂的一般公式开始计算 a_{12} 要容易的多.

例 1 排列

求把 n 个不同对象排列成一行的不同方法数的递推关系. 求 8 个对象的排列数.

设 a_n 表示 n 个不同对象的排列数. 对于这一行中第一个对象有 n 种选择. 其余 $n-1$ 个对象的任意排列可以跟随在这一选择之后, 即其余 $n-1$ 个对象的 a_{n-1} 个排列跟随其后. 因此 $a_n = na_{n-1}$. 在这个关系中递推地替换, 我们看到

$$a_n = na_{n-1} = n[(n-1)a_{n-2}] = \cdots = n(n-1)(n-2)\cdots \times 2 \times 1 = n!$$

特别地, a_8=8!. □

当然, 我们已知道 n 个对象的排列数是 $n!$. 在使用新技巧解决新问题之前, 用它来解决一个老问题, 看一下它是如何工作的, 这对我们很有帮助. 下面是一些例子.

例 2 爬台阶

一个小精灵要爬有 n 个台阶的楼梯. 它每迈一步能登上一个台阶或两个台阶. 求这个小精灵登上这个有 n 个台阶的楼梯的方法数 a_n 的递推关系.

容易验证, $a_1 = 1, a_2$=2(两个一步或者一个两步), a_3=3. 对于 $n = 4$, 图 7.1a 描绘出这个小精灵爬 4 个台阶的一种方法, 步伐的大小依次为 1, 2, 1. 其他可能的步伐是相继取大小为 2, 1, 1 的步子, 1, 1, 2 的步子, 2, 2 的步子, 或者 1, 1, 1, 1 的步子. 总之, 我们计数了 5 种爬 4 个台阶的方法; 所以 a_4=5.

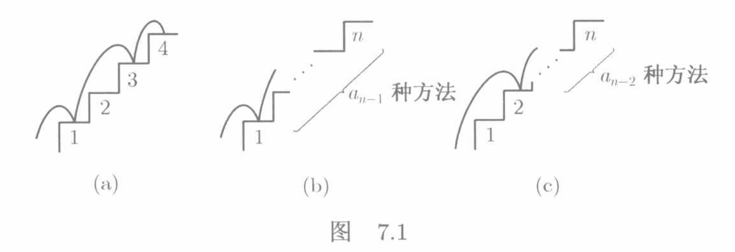


图 7.1

有没有某种系统的方法来列举爬 4 个台阶的方法, 使得我们可以把这个问题分解成由爬 3 个台阶的方法或者更少台阶的方法的部分呢? 显然, 一旦迈出第一步, 剩下要爬的台阶就是 3 个或者更少. 因此, 我们看到迈出一个台阶的第一步之后, 就有 a_3 种方法继续爬剩余的 3 个台阶. 如果第一步是跨上两个台阶, 那么有 a_2 种方法继续爬剩余的两个台阶. 所以 $a_4 = a_3 + a_2$. 我们断定, a_4, a_3, a_2 的值满足这个关系: 5=3+2. 这一讨论同样适用于爬任意数量的台阶时的第一步, 如图 7.1b 和图 7.1c 所示. 因此 $a_n = a_{n-1} + a_{n-2}$. □

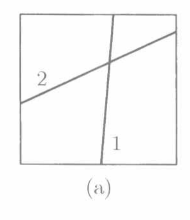
在 7.3 节, 我们得到上面这一递推关系的一个显式解. 我们称这一递推关系 $a_n = a_{n-1} + a_{n-2}$ 为**斐波那契关系**(Fibonacci relation). 由斐波那契关系以及初始

条件 $a_0 = a_1=1$ 所得到的数 a_n 称为**斐波那契数**(Fibonacci number). 前几个斐波那契数是 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89. 斐波那契数出现在组合数学的很多领域. 甚至有一本专门研究斐波那契关系和斐波那契数的杂志 *Fibonacci Quarterly*. 斐波那契数被运用于数学的其他领域, 例如数值分析等. 它还出现于自然界, 例如, 某些花的花瓣的排列. 关于出现在自然界的斐波那契数的相关信息参见 [1].

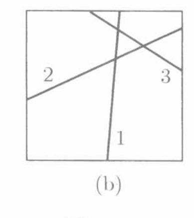
例 3　平面划分

假设我们在一张纸上画 n 条直线使得每一对直线相交 (但没有 3 条直线相交于同一点). 这 n 条直线可以把这个平面划分成多少个区域?

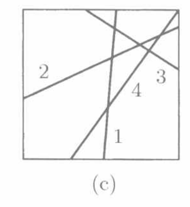
我们还是先通过对较小的 n 值分析这一状况来接近这个问题. 使用 1 条直线, 这张纸被分成 2 个部分. 使用 2 条直线, 我们得到 4 个区域, 即 $a_2=4$. 参见图 7.2a. 从图 7.2b, 我们看到 $a_3=7$. 好奇的读者可能会问, 怎样才能知道 3 条相交的直线总是可以创造 7 个区域? 那么让我们退回去一步.



(a)



(b)



(c)

图　7.2

显然, 两条相交直线总产生 4 个区域, 如图 7.2a 所示. 现在, 我们查看一下画出第 3 条直线 (在图 7.2b 中被标上 "3" 的那条直线) 的效应. 第 3 条直线必须与前面两条直线的每一条相交 (相交于不同点). 第 3 条直线在其与前 2 条直线形成的 2 个交点的前方、中间和后方分别跨越前 2 条直线所形成的 3 个区域 (第 3 条直线的效应不依赖于它是如何画的, 只依赖于它与前 2 条直线是如何相交的). 所以, 因为跨越了原来的 3 个区域, 第 3 条直线形成 3 个新区域, 实际上, 就是从原来 3 个老区域创建出 6 个新区域. 因此 $a_3 = a_2+3=4+3=7$, 这个值与第 3 条直线的画法无关.

类似地, 第 4 条直线在它与前 3 条直线形成的 3 个交点的前方、中间和最后交点的后方切割出 4 个区域 (参见图 7.2c), 所以 $a_4 = a_3 + 4 = 7 + 4 = 11$. 一般地, 第 n 条直线必须在它与前 $n-1$ 条直线形成的 $n-1$ 个交点的前方、中间和后方切割出 n 个区域. 所以, $a_n = a_{n-1} + n$. □

例 4　汉诺塔

汉诺塔是这样的一个游戏, 它有 n 个大小不一的圆环以及圆环要套入的 3 个柱子. 开始时, 所有圆环都放在第 1 个 (最左边) 柱子上, 而且最大的圆环在最底下, 其上依次是更小的圆环. 参见图 7.3a. 通过在柱子间移动圆环, 我们想要在第 3 根 (最右边的) 柱子上形成一个类似的锥形圆盘堆. 使问题变得复杂的是, 每当把一个

圆环移到一个新柱子上时, 这个被移动的圆环必须比这根新柱子上已有的所有圆环都小; 或者说, 在这个游戏的每一个阶段, 在每一个柱子上都一定是一个锥形.

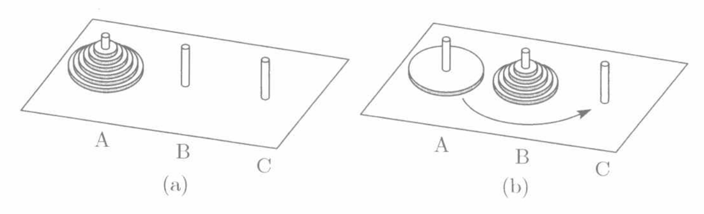


图 7.3 汉诺塔

求 a_n 的递推关系, 其中 a_n 是进行 n 个圆环的汉诺塔游戏时所需的最少移动次数. 玩 6 个圆环的游戏需要移动多少次? 在阅读我们的解决方案之前, 尝试着用一枚一角硬币、一枚一分硬币、一枚五分硬币和一枚二角五分硬币玩一玩这一 (4 个圆环的) 游戏.

对这一问题的关键观察是, 比如说在柱 A 上有 6 个小圆环, 我们想要把它们移动到柱 C 上, 那么首先必须 "玩 5 环汉诺塔游戏" 以便把柱子 A 上的 5 个小环移动到柱子 B 上, 然后把第 6 小的环从 A 移动到 C, 随后再在柱子 B 和 C 之间 "玩 5 环游戏", 参见图 7.3. (当然, 为了移动 5 个小环, 我们必须把第 5 小的环从 A 移动到 B, 这意味着从 A 到 C, 从 C 到 B 玩 4 环游戏, 以此类推.) 因此, 为了把 n 个圆环从 A 移动到 C, 必须首先把 $n-1$ 个小环从 A 移动到 B, 然后, 把最大的环 (第 n 个环) 从 A 移到 C, 然后, 把 $n-1$ 个小环从 B 移动到 C.

如果 a_n 是把 n 个圆环的锥形从一个柱子移动到另一柱子时所需的移动次数, 那么前面的描述产生下面的递推关系 $a_n = a_{n-1} + 1 + a_{n-1} = 2a_{n-1} + 1$. 初始条件是 a_1=1, 所以 $a_2 = 2a_1 + 1 = 3, a_3 = 2a_2 + 1 = 7, a_4 = 2a_3 + 1 = 15, a_5 = 2a_4 + 1 = 31, a_6 = 2a_5 + 1 = 63$. 所以, 6 环游戏需要 63 次移动. 注意, a_n 还满足公式 $a_n = 2^n - 1$. □

例 5 现金账户中钱的增长

一家银行每年对现金账户中的现金支付 4%的利息. 对于使用下面的投资策略, 求经过 n 年后, 账本上现金总数的递推关系.

(a) 投资 1 000 美元, 并把它放入这家银行 n 年.

(b) 在每年的年末投资 100 美元.

如果在一年的开始这个账户上有 x 美元, 那么这一年的年末 (即下一年的开始), 它应该有 x 美元再加上这 x 美元产生的利息, 倘若这一年中没有加钱也没有取钱. 对于部分 (a), 递推关系是 $a_n = a_{n-1} + 0.4a_{n-1} = 1.04a_{n-1}$. 初始条件 a_0= 1 000. 对于部分 (b), 递推关系必须反映出追加的 100 美元 (这 100 美元没赚到利息, 因为它是在年末加入的). 所以 $a_n = 1.04a_{n-1} + 100, a_0$=0. □

例 6 兑换

一连几天分发一片口香糖 (价值 1 美分), 或一根棒棒糖 (价值 10 美分), 或一个面包圈 (价值 20 美分), 直到价值 n 美分的食品发完为止, 求这样做的不同方法数的递推关系.

可以把这一问题与例 2 中的登台阶问题作类似的处理. 即, 如果第一天分发了价值 1 美分的口香糖, 那么还剩下 $(n-1)$ 美分的食品在随后的几天中分发; 如果第一天分发的是 10 美分的棒棒糖, 那么第二天有 $(n-10)$ 美分的食品分发; 如果第一天分发的是 20 美分, 那么接下来的几天有 $(n-20)$ 美元要分发. 所以 $a_n = a_{n-1} + a_{n-10} + a_{n-20}, a_0=1$(有一种方法分发 0 样东西), 显然对于 $k < 0$ 有 $a_k=0$. □

例 7　禁止的子序列

求没有子序列 “012” 出现的 n 位三进制序列的个数 a_n 的递推关系.

回想一下, 三进制序列是由若干个 0, 1, 2 组成的. 我们利用小精灵爬台阶问题所使用的分析来解决这一问题. 如果 n 位三进制序列的第一个数字是 1, 那么将有 a_{n-1} 个没有 012 模式的 $n-1$ 位三进制序列跟在这个初始 1 的后面. 如果第一位是 2, 情况类似.

然而, 如果第一位是 0, 就会出现一个问题. 跟在这个初始 0 之后没有模块 012 的 $n-1$ 位三进制序列可以是以 12 开始的序列. 尽管这样的 $(n-1)$ 位序列不包含模块 012, 但是 0 后面的这样的 $(n-1)$ 位序列是以 012 开始的 n 位序列. 通过减去所有以 012 开始的 (而在后面不出现子序列 012 的) n 位序列来改正这一错误. 这样的序列是由 012 后面跟着任意没有 012 模式的 $(n-3)$ 位三进制序列组成的. 有 a_{n-3} 个这样的序列. 因此, 要求的递推关系是: $a_n = a_{n-1} + a_{n-1} + (a_{n-1} - a_{n-3}) = 3a_{n-1} - a_{n-3}$. □

前面的几个例子探讨了递推关系模型, 它们不外乎采用两种方式, 或者把一个问题分解成第一步, 随后是一个更小集合的同样问题 (例 1, 例 2, 例 6, 例 7), 或者通过观察从 $n-1$ 的情况行进到 n 的情况时所发生的改变 (例 3 和例 5). 而例 4 则对上面所说的过程略有改动, 它是移动第 n 个圆环之前和之后是 $n-1$ 个圆环游戏.

有两种求解目前我们看到的一些关系的简单方法. 第一种方法是后向递推替换: 无论 a_{n-1} 在 a_n 的关系式的什么地方出现, 我们都用 a_{n-1}(包含 a_{n-2}) 的公式替换 a_{n-1}, 然后替换 a_{n-2}, 以此类推. 在例 3 中, 关系 $a_n = a_{n-1} + n$ 变成 $a_n = (a_{n-2} + n - 1) + n = \cdots = 1 + 2 + 3 + \cdots + n - 1 + n$. 我们在例 1 中使用这一方法得到 $a_n = n(n-1)(n-2) \times \cdots \times 2 \times 1$.

第二种方法是猜测这个递推关系的结果, 然后用数学归纳法加以证明. 在例 4 中, 我们注意到, 对递推关系 $a_n = 2a_{n-1} + 1, a_1=1$ 的前 6 个值来说有 $a_n = 2^n - 1$. 我们如下用数学归纳证明 $a_n = 2^n - 1$. 对于 $n = 1$, 成立. 假设 $a_{n-1} = 2^{n-1} - 1$, 那么我们有 $a_n = 2a_{n-1} + 1 = 2(2^{n-1} - 1) + 1 = 2^n - 2 + 1 = 2^n - 1$.

现在, 考虑涉及两个变量关系或联立关系的更复杂的递推关系.

例 8 无重复选取

设 $a_{n,k}$ 表示从 n 个不同对象的集合中选出 k 个对象的子集的方法数. 求 $a_{n,k}$ 的递推关系.

观察发现, $a_{n,k}$ 就是$\binom{n}{k}$. 依据是否使用第一个对象, 我们把这一问题分解成两个子情况. 有 $a_{n-1,k}$ 个不使用第一个对象的 k 对象子集. 有 $a_{n-1,k-1}$ 个使用第一个对象的 k 对象子集. 所以, 我们有 $a_{n,k} = a_{n-1,k} + a_{n-1,k-1}$. 这是帕斯卡三角恒等式 [5.5 节的等式 (3)]. 初始条件是对于所有的 $n \geqslant 0$ $(a_{n,k} = 0, k > n)$ 来说 $a_{n,0} = a_{n,n}$=1. □

例 9 分配

求把 n 个相同的球分配到 k 个不同盒子里且每个盒子里的球数在 2~4 的方法数的递推关系. 用 3 种颜色的球重复这一问题.

根据前面例子的要点, 我们考虑有多少个球进入到第 1 个盒子里. 如果在第 1 个盒子里放 2 个球 (有 1 种完成方法), 那么有 $a_{n-2,k-1}$ 种方法把其余的 $n-2$ 个相同球放入到剩下的 $k-1$ 个盒子里. 继续这一推理, 我们看到 $a_{n,k} = a_{n-2,k-1} + a_{n-3,k-1} + a_{n-4,k-1}$. 初始条件是 $a_{2,1} = a_{3,1} = a_{4,1} = 1, a_{n,1} = 0(n > 4)$.

如果现在有 3 种颜色球, 则有 C(2+3-1, 2)=6 种方法从 3 种颜色的球中重复地为第 1 个盒子选出 2 个球的子集. (回想一下 5.3 节开始处对重复选取的讨论.) 类似地, 有 C(3+3-1, 3)=10 种方法从 3 种类型的球中选出 3 个球, $C(4+3-1,4) = 15$ 种方法从 3 种类型中选出 4 个球. 于是 $a_{n,k} = 6a_{n-2,k-1} + 10a_{n-3,k-1} + 15a_{n-4,k-1}$ 且 $a_{2,1} = 6, a_{3,1} = 10, a_{4,1} = 15, a_{n,1} = 0$.

可以用生成函数解决所有球都相同时的问题 (参见 6.1 节例 3), 但是当附加限制是球的类型不同时, 递推关系是唯一实用的方法. □

例 10 放置括号

给在计算器上计算 n 个数的乘积 $k_1 \times k_2 \times k_3 \times k_4 \times \cdots \times k_n$ 放置括号, 括号的不同放置方法数目为 a_n, 求 a_n 的递推关系.

为了清楚地说明这一问题, 观察发现, 当两个数相乘时, 有一种放置括号的方法: $(k_1 \times k_2)$; 所以 a_2=1. 当 3 个数相乘 $k_1 \times k_2 \times k_3$ 时, 则有两个放置括号的方法, 即, $[(k_1 \times k_2) \times k_3]$ 和 $[k_1 \times (k_2 \times k_3)]$; 所以 a_3=2. a_0 和 a_1 应该是什么, 就没有那么显然了, 但是为了使最终的递推关系有一个简单的形式, 我们设 $a_0 = 0$ 和 a_1=1. 为了求 a_n 的递推关系, 我们查看一下 n 个数的积中最后的一次乘法 (最外面的括号). 最后的乘法包含两个乘法子问题的积:

$$(k_1 \times k_2 \times \cdots \times k_i) \times (k_{i+1} \times k_{i+2} \times \cdots \times k_n)$$

其中 i 的取值范围是 1 到 $n-1$. 分别给两个子问题放置括号的方法数是 a_i 和 a_{n-i}, 所以有 $a_i a_{n-i}$ 种方法给两个子问题都放置括号. 对所有 i 求和, 我们得到递推关系

$(n \geqslant 2)$ $a_n = a_1 a_{n-1} + a_2 a_{n-2} + \cdots + a_{n-1} a_1$. □

例 11　递推关系系统

求下面问题的递推关系:

(a) 有偶数个 0 的 n 位二进制序列数;

(b) 有偶数个 0 和偶数个 1 的 n 位三进制序列数.

(a) 我们使用爬台阶模型的第一步分析, 求有偶数个 0 的 n 位二进制序列个数 a_n 的递推关系. 如果 n 位二进制序列以 1 开始, 那么需要在剩余的 $(n-1)$ 位二进制序列中有偶数个 0, 有 a_{n-1} 个这样的序列. 如果这个 n 位二进制序列以 0 开始, 那么需要在剩余的 $n-1$ 位二进制序列中有奇数个 0, 有 $2^{n-1} - a_{n-1}$ 个这样的序列, 因为所有 2^{n-1} 个 $(n-1)$ 位序列减去有偶数个 0 的 $(n-1)$ 位二进制序列得到有奇数个 0 的 $(n-1)$ 位二进制序列. 总之, $a_n = a_{n-1} + (2^{n-1} - a_{n-1}) = 2^{n-1}$. 我们的递推关系简化成一个 a_n 的公式. 换句话说, 每一个 $(n-1)$ 位序列可以产生一个有偶数个 0 的序列, 其方法是: 如果这个 $(n-1)$ 位序列有偶数个 0, 则在这个序列的前面加上一个 1; 如果这个序列有奇数个 0, 则在这个序列前面加上一个 0.

(b) 设 a_n 是有偶数个 0 且有偶数个 1 的 n 位三进制序列数, b_n 是有偶数个 0 和奇数个 1 的 n 位三进制序列数, c_n 是有奇数个 0 和偶数个 1 的 n 位三元序列数, 我们要求 a_n, b_n, c_n 的联立递推关系. 观察发现, $3^n - a_n - b_n - c_n$ 是有奇数个 0 和奇数个 1 的 n 位三进制序列的个数. 一个有偶数个 0 和偶数个 1 的 n 位三进制序列可以通过下面的方法得到: 让一个 1 做这个序列的第一位, 后面跟着一个有偶数个 0 和奇数个 1 的 $(n-1)$ 位三进制序列; 或者一个 0 后面跟着一个有奇数个 0 和偶数个 1 的一个 $(n-1)$ 位序列; 或者一个 2 后面跟着一个有偶数个 0 和偶数个 1 的 $(n-1)$ 位序列. 因此, $a_n = b_{n-1} + c_{n-1} + a_{n-1}$. 通过类似的分析, 可以得到 $b_n = a_{n-1} + (3^{n-1} - a_{n-1} - b_{n-1} - c_{n-1}) + b_{n-1} = 3^{n-1} - c_{n-1}$ 和 $c_n = a_{n-1} + (3^{n-1} - a_{n-1} - b_{n-1} - c_{n-1}) + c_{n-1} = 3^{n-1} - b_{n-1}$. 初始条件是 $a_1 = b_1 = c_1 = 1$. 为了递推地计算 a_n 的值, 必须同时计算 b_n 和 c_n. □

本节最后介绍一些关于差分方程的内容. 序列 $(a_0, a_1, a_2, \cdots)$ 的**一阶 (反向) 差分**[first (backward) difference]Δa_n 定义为 $\Delta a_n = a_n - a_{n-1}$. 二阶差分是 $\Delta^2 a_n = \Delta a_n - \Delta a_{n-1} = a_n - 2a_{n-1} + a_{n-2}$, 以此类推. 差分方程是包含 a_n 及其差分的方程, 例如 $2\Delta^2 a_n - 3\Delta a_n + a_n = 0$. 观察发现:

$$
\begin{aligned}
&a_{n-1} = a_n - (a_n - a_{n-1}) = a_n - \Delta a_n \\
&a_{n-2} = a_{n-1} - \Delta a_{n-1} = (a_n - \Delta a_n) - \Delta(a_n - \Delta a_n) = a_n - 2\Delta a_n + \Delta^2 a_n (*)
\end{aligned}
$$

可以通过类似的方程用 $a_n, \Delta a_n, \cdots, \Delta^k a_n$ 表示 a_{n-k}. 因此, 任意递推关系可以改写成差分方程的形式, 即用 a_n 及其差分来表示 (*) 右边的 a_{n-k}. 反过来, 通过把 Δa_n 写成 $a_n - a_{n-1}$, 等, 任意的差分方程可以写成递推关系.

在计算机上对微分方程求解时, 通常使用差分方程来逼近微分方程. 差分方程本身作为动态系统的模型而有着广泛的应用, 对此, (需要连续函数的) 微分方程是无法比拟的. 在经济学中, 我们使用差分方程模拟在连续的几年内国民总产值的预测. 在生态学中, 我们使用差分方程模拟各物种在连续几年内的数量. 正如上面所提示的那样, 任意差分方程模型还可以像公式一样化成一个递推关系. 然而, 动态系统的行为更容易用差分来分析和解释. 关于差分方程及其应用的更多信息, 参见 Sandefur[4].

例 12　两动物物种模型

假设如果没有狐狸的干扰, 兔子数量每年的增量是 αr_n, 其中 r_n 是兔子的数量. 但是当狐狸出现时, 每只兔子被狐狸吃掉的概率是 βf_n (f_n 是狐狸数). 狐狸每年只减少 γf_n, 但是当兔子出现时, 每一只狐狸存活并生育一只小狐狸的概率是 δf_n(狐狸的死亡已在 γf_n 项中考虑). 给出一个联立差分方程来描述在连续几年内狐狸和兔子的数量.

上面给出的两个物种在数量上的年度改变信息可以产生如下差分方程:

$$\Delta r_n = \alpha r_n - \beta r_n f_n$$

$$\Delta f_n = -\gamma f_n + \delta r_n f_n$$

□

练　习　7.1

练习概况　前 39 道练习要求的都是类似于例子中的递推关系模型. 练习 28 到练习 40 需要多重下标和方程, 练习 41~43 是关于差分方程的练习. 其余的练习则是层次更高的练习.

1. 求把 n 个不同的对象分配到 4 个盒子里的方法数的递推关系. 初始条件是什么?

2. **(a)** 设例 2 中的小精灵每一步可以登上 1 个、2 个或 4 个台阶, 求它爬 n 个台阶的方法数的递推关系?

(b) 这个小精灵有多少种爬 4 个台阶的方法?

3. 设我们能使用的车种是卡迪拉克、悍马和福特. 一辆悍马需要两个车位, 而卡迪拉克或福特只需要一个车位. 求在 n 个车位上把汽车排成一行的方法数的递推关系.

4. **(a)** 假设每小时行进的方式有 3 种, 或者以每小时 3 英里速度步行, 或者以每小时 5 英里速度慢跑, 或者以每小时 10 英里的速度跑步, 每一小时结束时, 要选择下一个小时的行进方式, 求行进 n 英里的方法数的递推关系.

(b) 有多少种方法行进 12 英里?

5. 在连续几天内使用 1 958 枚 1 美分、1 971 枚 1 美分、1 951 枚 5 美分、1 967 枚 5 美分、1 959 枚 10 美分和 1 975 枚 25 美分来发放总额为 n 美分的钱, 求发放方法数的递推关系.

6. **(a)** 求没有两个连续 1 出现的 n 位二进制序列个数的递推关系.

(b) 对 n 位三进制序列, 重复上面的问题.

(c) 对没有连续 1 或连续 2 的 n 位三进制序列, 重做问题 (a).

7. 求 n 个月后兔子对数的递推关系, 如果: (1) 最初有一对刚刚出生的兔子; (2) 每过一个月, 年龄超过一个月的每一对兔子都要繁殖一对后代 (一雄一雌).

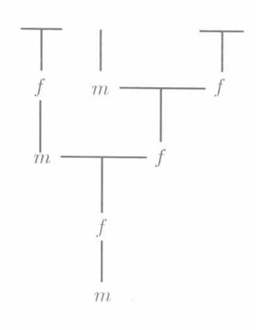
8. 证明: 二项式和



$$s_n = \binom{n+1}{0} + \binom{n}{1} + \binom{n-1}{2} + \cdots$$

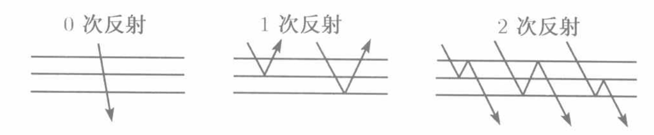
满足斐波那契关系.

9. 求把 n 个多米诺置入一个 $2 \times n$ 棋盘的方法数的递推关系.



10. 设雄蜂是雌蜂通过无性繁殖而来的, 而雌蜂有常规的雄性和雌性父母. 求一只雄性蜜蜂的第 n 代前辈的蜜蜂数量 a_n 的递推关系. 左图的祖先图表显示 $a_1 = 1, a_2 = 2, a_3$=3.

11. 有两块邻接的玻璃, 求一个图像经由这两块玻璃反射 n 次的方法数 a_n 的递推关系. 下面的图表示 $a_0 = 1, a_1 = 2, a_2$=3.



12. 求在一张纸上 n 个两两相交的圆所创建的区域数的递推关系 (没有 3 个圆有相同的交点).

13. **(a)** 设一张纸上的 n 条直线中有 k 条是平行的, 而其余的 $n-k$ 条直线都与其他直线相交 (但是没有 3 条直线相交于一点), 求由这 n 条直线创建的区域数的递推关系.

(b) 对于 $n = 9, k = 3$, 求区域数.

14. 展示下列各汉诺塔游戏规则的工作情况.

(a) 在奇数次移动, 按顺时针方向把最小圆环移动到下一个圆柱上 (把 3 根圆柱想象成是三角形的 3 个角点); 在偶数次移动, 做不使用最小圆环的唯一合法移动.

(b) 按从小到大的顺序用 1 到 n 对圆环编号. 从不连续移动同一个圆环两次. 总是把编号为偶数的圆环放在编号为奇数圆环的上面 (或一根空柱子上), 并把编号为奇数的圆环放在编号为偶数的圆环的上面 (或一根空柱子上).

15. 设利息是 6%且在每一年的年初追加 50 美元, 求经过 n 年后账户中的钱数的递推关系.

16. **(a)** 设利息是 8%, 且年还贷额是 3 000 美元 (在每年年末计算利息之后还贷), 求贷款 30 000 美元, 经过 n 年后总还贷金额的递推关系.

(b) 使用计算器或者计算机, 确定需要多少年才能还贷完所有贷款?

17. Angela 每天在熟食店吃午餐, 预订下面食品之一: 鸡丁沙拉、金枪鱼三明治或火鸡卷. 如果她从不连续 3 天预订鸡丁沙拉, 求她 n 天订午餐的花样数的递推关系.

18. 使用字母 A, B, C 做 n 字母序列, 使得不在这个序列最后位置上的任意 A 的后面总是跟着一个 B. 求这样的 n 字母序列数目的递推关系.

19. **(a)** 求使用若干个 1、若干个 3 和若干个 5 组成的、和为 n 的项的序列个数的递推关系.

(b) 对 5 后面不能跟着 1 的附加条件, 重做 (a).

(c) 对没有子序列 135 的附加条件, 重做 (a).

20. **(a)** 求在一个高 n 英尺的旗杆上排列 3 种类型的旗子的方法数的递推关系：红旗 (高 1 英尺)、金旗 (高 1 英尺)、绿旗 (高 2 英尺).

(b) 对没有 3 面 1 英尺高的旗帜连在一起的附加条件, 重做 (a).

(c) 对没有红旗在金旗之上, 而这个金旗又在绿旗之上的附加条件, 重做 (a).

21. 设在 n 天, 每天送出 1 美元、2 美元或 3 美元, 且送出 1 美元的天数是偶数天, 求送出这些钱的方法数 a_n 的递推关系.

22. 求计数至少有一组连续的 0 的 n 位二进制序列的数目的递推关系.

23. 求至少有一个 1 且第一个 1 一定出现在第一个 0 之前 (可能没有 0) 的 n 位四进制 (0, 1, 2, 3) 序列的数目的递推关系.

24. 求 1 不出现在 2 的右边的 n 位三进制 (0, 1, 2) 序列的数目的递推关系.

25. 求模式 "012" 第一次出现是在序列末端的 n 位三进制 (0, 1, 2) 序列的数目的递推关系.

26. 一个开关游戏有 n 个开关, 开始时所有开关都处于 OFF 状态. 为了能够切换第 i 个开关, 第 $(i-1)$ 个开关必须是 ON, 而更前面的所有开关必须都是 OFF. 第一个开关总是可切换的. 求为使第 n 个开关是 ON 而其他所有开关是 OFF, 这 n 个开关必须弹起的总次数的递推关系.

27. 求给 $2n$ 个人配对进行网球比赛的方法数的递推关系.

28. 求从 n 个类型的对象中重复地选出 k 个对象的方法数的递推关系.

29. 求从 k 个类型的对象中选出 n 个对象使得每种类型最多有 3 个对象的方法数的递推关系.

30. 设从 k 个不同类型的面包圈中预定 n 个面包圈, 且每种类型必须选出 2 个、4 个或 6 个面包圈, 求这样预定的数目 $a_{n,k}$ 的递推关系.

31. 求把整数 n 分成 k 个部分的分拆数的递推关系.

32. 把 n 个相同的对象分配到 k 个不同的盒子里且每个盒子至多有 4 个对象, 而且正好有 m 个盒子有 4 个对象, 求这一分配数 $a_{n,m,k}$ 的递推关系.

33. 求计算有偶数个 0 和偶数个 1 的 n 位二进制序列的数目的递推关系系统.

34. 求计算符合下列条件的 n 位四进制序列的数目的递推关系系统.

(a) 偶数个 0. **(b)** 0 和 1 的总数是偶数个. **(c)** 偶数个 0 和偶数个 1.

35. 求计算正好有一对连续 0 的 n 位二进制序列的数目的递推关系系统.

36. 求计算有 k 对连续 1 而没有连续 0 的 n 位二进制序列的数目的递推关系系统.

37. 连续几天施舍 1 美分、5 美分或者 10 美分, 直到给出 n 分, 使得不能在连续 2 天施舍相同数量的钱, 求计算施舍方法数的递推关系系统.

38. 求把圆上 $2n$ 个点配对且没有相交弦的方法数的递推关系. (提示：如例 11 所示, 递推关系包含 a_k 的乘积.)

39. 求把 n 凸边形分成三角形且没有相交对角线的配对方法数的递推关系.

40. 求有 n 个标签叶子的二叉树个数的递推关系.

41. 对下面各式所给的 a_n, 求 Δa_n 和 $\Delta^2 a_n$.

(a) $3n+2$. **(b)** n^2. **(c)** n^3.

42. 设 f_n 是用 n 美元可以买到的食品量. 设 p_n 是 n 美元食品的 "估测值". 假设估测值增加 1 美元的食品量等于实际食品量增加的百分比. 求关于 Δp_n 和 Δf_n 的差分方程.

43. **(a)** 求前 n 个整数的置换中, 每个整数与置换中它左边的某个整数相差 1(第一个数除外) 的置换数的递推关系. 初始条件是什么?

(b) 通过猜测给出 (a) 的递推关系, 并用归纳法证明它.

(c) 对这一问题直接给出组合答案.

44. (a) 求 1 到 n 中任意两个整数都不连续的 k 个整数的子集的数目 $f(n,k)$ 的递推关系.

(b) 证明: $\sum\limits_{k=0}^{n/2} f(n,k) = F_{n+1}$, 其中 F_{n+1} 是第 $(n+1)$ 个斐波那契数 (n 是偶数), 其中 $F_0 = F_1 = 1$.

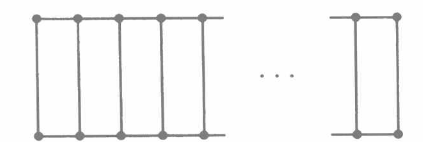
45. 设价值等于 n 美分的相同的 1 美分、相同的 5 美分、相同的 10 美分和相同的 25 美分的组合数为 a_n, 求计算 a_n 的递推关系.

46. 求投掷一枚硬币 $2n$ 次, 有 n 个正面和 n 个反面且有下面各限制的方法数的递推关系:

(a) 在任意一次投掷时, 正面数都不小于反面数.

(b) 只有经 $2n$ 次投掷后, 正面数才等于反面数.

47. 求周长为 n 的不同整数面积三角形的数目的递推关系 (这个关系因 n 是奇数还是偶数而不同).



48. 求计算有 $2n$ 个顶点的 "梯子" 图的生成树的数量的递推关系系统.

49. 用数学归纳法或组合论述证明下面斐波那契数的等式 (F_i 是第 i 个斐波那契数). 这里 $F_0 = F_1 = 1$.

(a) $\sum\limits_{i=0}^{n} F_i = F_{n+2} - 1$. (b) $\sum\limits_{i=0}^{n} F_i^2 = F_n F_{n+1}$.

(c) $\sum\limits_{k=0}^{n} F_{2k} = F_{2n+1}$. (d) $F_n F_{n+2} = F_{n+1}^2 + (-1)^n$.

(e) $F_1 - F_2 + F_3 - \cdots - F_{2n} = -F_{2n-1}$.

50. (a) 证明: $F_{n+m} = F_m F_n + F_{m-1} F_{n-1}$.

(b) 根据部分 (a) 得出 F_{n-1} 整除 F_{kn-1} 的结论.

7.2 分治关系

本节给出一类特殊的递推关系, 它们通常源自于计算机递归算法的分析中. 这些算法使用 "分治" 方法递归地把一个问题分解成一半大小的两个子问题. 3.1 节例 3 中的词典搜索就是一个这样的算法. 二叉树也直接或间接地与大多数 "分治" 算法有关. 反过来, 利用分治递推关系很容易解决很多有关树 (第 3 章的主题) 的计数问题.

利用分治算法处理 n 元素问题所需的步数 a_n 通常满足下面这种形式的递推关系:

$$a_n = ca_{n/2} + f(n) \tag{1}$$

下面的表格对于 c 和 $f(n)$ 一些常用值给出 (1) 的解的形式 ($\lceil r \rceil$ 是满足 $m \geqslant r$ 的最小整数 m):

c	$f(n)$	a_n
$c=1$	d	$d\lceil\log_2 n\rceil$+A
$c=2$	d	$An-d$
$c>2$	dn	$An^{\log_2 c}+\left(\frac{2d}{2-c}\right)n$
$c=2$	dn	$dn(\lceil\log_2 n\rceil+A)$

我们要选择常数 A 以满足初始条件.

如果能够把一个问题递归地分解成 k 个部分, 而不是两个部分, 那么就应该在前面表格的所有地方用 k 取代 2, 只有 $c=k$ 除外, 而 $f(n)=d$ 变成 $An-d/(k-1)$. 例如, 递推关系

$$a_n = ca_{n/k}+dn, \quad c\neq k$$

有下面的解:

$$a_n = An^{\log_k c}+\left(\frac{kd}{k-c}\right)n \tag{2}$$

通过替换很容易验证前面表格中给出的 a_n 的解. 考虑这样的情况: $a_n = ca_{n/2}+dn, c\neq 2$. 我们把表格中的 a_n=$An^{\log_2 c}+[2d/(2-c)]n$ 代入到 $ca_{n/2}+dn$, 我们得到:

$$\begin{aligned}ca_{n/2}+dn &= c\left[A\left(\frac{n}{2}\right)^{\log_2 c}+\left(\frac{2d}{2-c}\right)\frac{n}{2}\right]+dn\\ &= \frac{cAn^{\log_2 c}}{2^{\log_2 c}}+\frac{cdn}{2-c}+\frac{(2-c)dn}{2-c}\\ &= \frac{cAn^{\log_2 c}}{c}+\frac{cdn+(2-c)dn}{2-c}=An^{\log_2 c}+\left(\frac{2d}{2-c}\right)n=a_n\end{aligned}$$

下面的例子给出这样的“分治”递推关系和它们的解.

例 1　联赛中的轮次

在一次网球联赛中, 每名选手在第一轮都要进行一场比赛. 接下来, 第一轮的所有胜者参加第二轮比赛. 胜者继续进入到下一轮, 直到最后只剩一名选手, 即这次联赛的冠军. 假设对于某个 k, 联赛有 $n=2^k$ 名选手, 寻求并解出有 n 名选手的联赛轮数的递推关系.

用二叉树表示的话, a_n 是有 $n=2^k$ 个叶子的平衡树的高度. 因为在每轮比赛后, 就有一半的选手被淘汰, 所以当选手数量变成 $\frac{1}{2}$ 时, 轮数就增加 1. 轮数 a_n 的递推关系是:

$$a_n = a_{n/2}+1$$

根据前面的表格, 这个递推关系的解是 $a_n=\log_2 n+A$. 为了确定 A, 观察发现 $0=a_1=\log_2 1+A=0+A$, 所以 $A=0$. □

例 2　求集合中的最大数和最小数

构建一个递推关系模型, 计数在下面的算法之下寻找 n 个不同整数的集合 S 中最大数 l 和最小数 s 所需的比较次数. 然后对这个递推关系求解.

开始时, 假定 n 是一个偶数. 假设我们已经在 S 的第一个一半 (这一半有 $n/2$ 个成员) 中找到最大数和最小数, 分别是 l_1 和 s_1, 并在 S 的第二个一半中找到了最大数和最小数, 分别是 l_2 和 s_2. 然后做两次比较, 一次是 l_1 与 l_2 之间的比较, 另一次是 s_1 与 s_2 之间的比较, 这样就找到了 S 中的最大数 l 和最小数 s.

在这样的过程中, 相应的比较次数的递推关系是 $a_n = 2a_{n/2}+2$, 其中 $n \geqslant 4$ 且是偶数. 如果 n 是奇数, 我们可以把 S 分成两个几乎完全相同的部分, 这个递推关系是 $a_n = a_{(n+1)/2} + a_{(n-1)/2} + 2$. 观察发现 a_1=0, 因为这个数既是最大又是最小的. 而且 a_2=1, 因为在一个两元素集合中, 我们可以用一次比较确定其最大数与最小数. 利用这两个递推关系以及 a_1 和 a_2, 我们可以递推地确定对于任意的 n 所需要的比较次数.

接下来, 我们求解递推关系 $a_n = 2a_{n/2} + 2$, 其中 $n \geqslant 4$ 且 a_2=1. 前面的表格告诉我们这个关系的解的形式是 $a_n = An - 2$. 把解 $a_n = An - 2$ 代入到递推关系 $a_n = 2a_{n/2} + 2$ 的两边, 得到:

$$An-2 = a_n = 2a_{n/2}+2 = 2\left(A\frac{n}{2}-2\right)+2 = An-4+2 = An-2$$

可以利用初始条件 a_2=1 来确定 A:

$$1 = a_2 = A_1(2) - 2 \quad \text{或} \quad A_1 = \frac{3}{2}$$

所以, $a_n = \frac{3}{2}n - 2$ 是依照前面给定的过程寻找最大数与最小数时所需的比较次数. (可以证明无法通过少于 $\frac{3}{2}n - 2$ 次比较来完成这一工作.) □

例 3　高效多位乘法

为了把两个 n 位数乘起来, 通常必须做 n^2 次位对位的乘法. 我们使用分治方法开发更快的算法.

先假设 n 是 2 的幂. 设 g 和 h 是两个 n 位数. 把这两个数都分成两个 $n/2$ 位的部分:

$$g = 10^{n/2}g_1 + g_2, \qquad h = 10^{n/2}h_1 + h_2$$

于是有

$$g \times h = (g_1 \times h_1)10^n + (g_1 \times h_2 + g_2 \times h_1)10^{n/2} + g_2 \times h_2 \tag{3}$$

观察发现

$$g_1 \times h_2 + g_2 \times h_1 = (g_1+g_2)\times(h_1+h_2) - g_1\times h_1 - g_2 \times h_2$$

所以, 我们只需要做 3 次 $n/2$ 位乘法, $g_1 \times h_1, g_2 \times h_2$ 和 $(g_1+g_2)\times(h_1+h_2)$ 来确定 (3) 中的 $g \times h$(实际上, (g_1+g_2) 和 (h_1+h_2) 可能是 $(n/2+1)$ 位数, 但是这一

点改变不影响方案的一般量级). 如果 a_n 代表根据前面的过程做两个 n 位数的乘积所需的位对位乘法的次数, 那么这个过程产生递推关系 $a_n = 3a_{n/2}$.

根据本节开始时给出的表格 (其中 $d = 0$), a_n 与 $n^{\log_2 3} \approx n^{1.6}$ 成正比, 这个结果比 n^2 有了显著的改进.

□

在某些情况下, 较之于 a_n 的精确公式, 我们对它的一般增长率更感兴趣. 下面的定理是由 Cormen, Leiserson, and Rivest[2] 给出的, 这个定理给出了这样的增长率的上界:

定理 设递推关系 $a_n = ca_{n/k} + f(n)$ 中 c 是正常数而 $f(n)$ 是正函数.

(a) 如果对于较大的 $n, f(n)$ 的增长与 $n^{\log_k c}$ 成正比 [即, 存在正常数 p 和 p', 使得 $pn^{\log_k r} \leqslant f(n) \leqslant p'n^{\log_k c}$], 那么 a_n 的增长率与 $n^{\log_k c}\log_2 c$ 成正比.

(b) 如果对于较大的 $n, f(n) \leqslant pn^q$, 其中 p 是正常数, 而且 $q < \log_k c$, 那么 a_n 的增长率与 $n^{\log_k c}$ 成正比.

练 习 7.2

1. 假设 n 是 2 的幂, 求解下面的递推关系 (留下待定的常数 A).

(a) $a_n = 2a_{n/2} + 5$. **(b)** $a_n = 2a_{n/4} + n$.

(c) $a_n = a_{n/2} + 2n - 1$. **(d)** $a_n = 3a_{n/3} + 4$.

(e) $a_n = 16a_{n/2} + 5n$. **(f)** $a_n = 4a_{n/2} + 3n$.

2. 寻求并求解有 n 名选手的联赛中要进行的比赛场次数的递推关系, 其中 n 是 2 的幂.

3. 在一家有 n 名销售人员的大公司里, 每 10 个人要向一名部门主管汇报, 每 10 名部门主管又要向一名区域主管汇报, 一直这样下去, 直到最后 10 名副总裁向一名公司总裁汇报. 如果这个公司有 n 名销售人员, 其中 n 是 10 的幂, 那么求解下面各数值的递推关系:

(a) 这个公司的不同主管层数. **(b)** 这个公司的主管人数 (一直到总裁).

4. 在一次网球联赛中, 每名选手都可获得 k 百美元, 其中 k 是这名选手在子联赛中所击败的人数 (这些人是这次联赛的子集, 它包含这名选手, 这名选手的手下败将, 以及他们的手下败将, 如此等等; 丢掉第一轮比赛的选手可以得到 100 美元). 如果这次联赛有 n 名选手, 其中 n 是 2 的幂, 求解这次联赛的奖金总额的递推关系.

5. 考虑下面以递增顺序排列 n 个不同整数 $x_1, x_2, \cdots, x_n$ 的方法 (n 是 2 的幂). 给这些整数配对 $\{x_1, x_2\}, \{x_3, x_4\}$, 等等. 比较每一对整数, 然后把较小的整数放在前面. 然后再把这些对配对成为四元素集合, 再合并这些有序对得到有序的四元组. 继续这一过程, 直到整个集合有序. 求解排列 n 个不同整数所需的比较次数的递推关系. (提示: 首先求合并两个有序 k 元组成为一个有序 $2k$ 元组所需的比较次数).

6. 在标准的淘汰赛中, 当一名选手在第 k 轮比赛中赢得比赛, 那么他或她将获得 $100k$ 美元 (例如, 第一轮获得 100 美元, 第二轮获得 200 美元). 求解 a_n 的递推关系, 其中 a_n 是在有 n 名选手的联赛中发放出去的奖金总额, 假设 n 是 2 的幂.

7. 用本节给出的解的形式替换下面各递推关系, 来验证它们.

(a) $a_n = a_{n/2} + d$. **(b)** $a_n = 2a_{n/2} + d$. **(c)** $a_n = 2a_{n/2} + dn$.

8. **(a)** 利用分治法, 设计一个求 n 个不同整数的集合中的最大数和第二大数的过程.

(b) 对于上面的过程, 给出它执行的比较次数的递推关系.

(c) 求解 (b) 所得到的递推关系.

9. **(a)** 使用分治法, 设计一个求 n 个不同整数的集合的最大数的过程.

(b) 给出该过程所执行的比较次数的递推关系.

(c) 求解 (b) 所得到的递推关系.

7.3 线性递推关系的解

本节说明如何求解下面形式的递推关系:

$$a_n = c_1 a_{n-1} + c_2 a_{n-2} + \cdots + c_r a_{n-r} \tag{1}$$

其中, c_i 是给定的常数. 有一个简单的技术来对这样的递推关系求解. 学习过常系数线性微分方程的读者将看到我们在这里研究的解的形式与他们的解非常相似. (1) 的一般解是形如 $a_n = \alpha^n$ 的各个解之和. 为了确定 α 是什么, 我们用 α^k 替换 (1) 中的 a_k, 于是得到

$$\alpha^n = c_1 \alpha^{n-1} + c_2 \alpha^{n-2} + \cdots + c_r \alpha^{n-r} \tag{2}$$

把上面等式的两边同时除以 α^{n-r} 来降低 (2) 中的所有项中 α 的幂:

$$\alpha^r = c_1 \alpha^{r-1} + c_2 \alpha^{r-2} + \cdots + c_r \tag{3}$$

或等价地,

$$\alpha^r - c_1 \alpha^{r-1} - c_2 \alpha^{r-2} - \cdots - c_r = 0 \tag{4}$$

等式 (4) 称为递推关系 (1) 的**特征方程**(characteristic equation). 它有 r 个根, 其中一些可能是复数 (但是开始时我们假设没有重根).

如果 $\alpha_1, \alpha_2, \cdots, \alpha_r$ 是 (4) 的 r 个根, 那么对于任意的 $i, 0 \leqslant i \leqslant r$, $a_n = \alpha_i^n$ 是递推关系 (1) 的解. 容易验证这些解的任意线性组合仍然是解 (练习 8). 即对于任意的常数 $A_i, 1 \leqslant i \leqslant r$, 下式是 (1) 的一个解:

$$a_n = A_1 \alpha_1^n + A_2 \alpha_2^n + \cdots + A_r \alpha_r^n \tag{5}$$

回想一下, 对于在右侧包含 $a_{n-1}, a_{n-2}, \cdots, a_{n-r}$ 的递推关系, 我们需要知道前 r 个值 $a_0, a_1, a_2, \cdots, a_{r-1}$ 的初始条件. 用 $a_1', a_2', \cdots, a_{r-1}'$ 表示这一组初始值. 于是我们必须选择满足这 r 个初始条件的 A_i:

$$A_1 \alpha_1^k + A_2 \alpha_2^k + \cdots + A_r \alpha_r^k = a_k', \quad 0 \leqslant k \leqslant r-1 \tag{6}$$

利用高斯消元法确定 r 个常量 A_i(记住, 在此阶段 α_i 是已知的), 我们就可以对 (6) 中的 r 个线性方程求解. 利用这些确定的 A_i, 就可以得到要求的 a_n 的解, 这个解

满足 (4), 满足递推关系 (1), 并满足初始条件 $a_0 = a'_0, a_1 = a'_1, \cdots, a_{r-1} = a'_{r-1}$. 如果特征方程 (4) 的解有 m 重根 α_*, 那么可以证明 $\alpha_*^n, n\alpha_*^n, n^2\alpha_*^n, \cdots, n^{(m-1)}\alpha_*^n$ 是用在 (5) 和 (6) 中的 m 个相关的解.

例 1　加倍兔子数量

每一年, Finch 博士的兔子数量都成倍增加. 他开始时有 6 只兔子. 经过 8 年后, 他有多少只兔子? 经过 n 年后呢?

如果 a_n 是经过 n 年后兔子的数量, 那么 a_n 满足关系 $a_n = 2a_{n-1}$. 我们已知 a_0=6. 做替换 $a_n = \alpha^n$, 我们得到 α^n=$2\alpha^{n-1}$, 或者, 两边除以 α^{n-1}, 得 α=2. 所以 a_n=2^n 是单一解, 而 $a_n = A2^n$ 是一般解. 初始条件是 6=$a_0 = A2^0$, 即 $A = 6$. 于是所求的解是 a_n=6×2^n. 经过 8 年后, 我们有 $a_8 = 6 \times 2^8 = 6 \times 256 = 1\ 536$ 只兔子. □

例 2　二阶线性递推关系

求解递推关系 $a_n = 2a_{n-1} + 3a_{n-2}$, 使得 $a_0 = a_1$=1.

设 $a_n = \alpha^n$, 我们得到特征方程:

$$\alpha^n = 2\alpha^{n-1} + 3\alpha^{n-2}$$

由此得:

$$\alpha^2 = 2\alpha + 3$$

上面的方程可以写成 $\alpha^2 - 2\alpha - 3$ =0 或者 $(\alpha - 3)(\alpha + 1) = 0$. 即, 根是 +3 和 −1. 所以, 这个递推关系的基本解是 $a_n = 3^n$ 和 $a_n = (-1)^n$, 而一般解是

$$a_n = A_1 3^n + A_2(-1)^n$$

现在使用下面的初始条件确定 A_1 和 A_2:

$$1 = a_0 = A_1 3^0 + A_2(-1)^0 = A_1 + A_2$$
$$1 = a_1 = A_1 3^1 + A_2(-1)^1 = 3A_1 - A_2$$

求解这两个联立方程, 得到 A_1=1/2, A_2=1/2(把这两个方程加起来消去 A_2 得到 $2 = 4A_1$, 或 A_1=1/2, 然后再确定 A_2). 有最初给定的初始条件的递推关系的解是:

$$a_n = \frac{1}{2} \times 3^n + \frac{1}{2} \times (-1)^n.$$ □

例 3　斐波那契关系的解

求 7.1 节例 2 中的小精灵爬上 n 个台阶的方法数的公式.

在 7.1 节例 2 中, 我们得到这个递推关系是 $a_n = a_{n-1} + a_{n-2}$, 且初始条件是 $a_1 = 1, a_2 = 2$, 或等价地, $a_0 = a_1$=1. 设 $a_n = \alpha^n$, 我们得到相应的特征方程:

$$\alpha^n = \alpha^{n-1} + \alpha^{n-2}$$

上面的方程可以简化成

$$\alpha^2=\alpha+1 \quad \text{或} \quad \alpha^2-\alpha-1=0$$

使用二次方程求解公式, 我们得到

$$\alpha=\frac{1}{2(1)}[-(-1)\pm\sqrt{(-1)^2-4(1)(-1)}]=\frac{1}{2}(1\pm\sqrt{5})$$

即, 我们有根 $\frac{1}{2}+\frac{1}{2}\sqrt{5}$ 和 $\frac{1}{2}-\frac{1}{2}\sqrt{5}$, 而这个问题的一般解是:

$$a_n=A_1\left(\frac{1}{2}+\frac{1}{2}\sqrt{5}\right)^n+A_2\left(\frac{1}{2}-\frac{1}{2}\sqrt{5}\right)^n$$

我们把证明 $A_1=\frac{1}{\sqrt{5}}(\frac{1+\sqrt{5}}{2})$ 和 $A_2=-\frac{1}{\sqrt{5}}(\frac{1-\sqrt{5}}{2})$ 留作练习. 我们注意到一个令人惊讶的事实: 为了生成斐波那契整数序列 $1,1,2,3,5,8,13,\cdots$, 我们需要 $(\frac{1}{2}+\frac{1}{2}\sqrt{5})$ 和 $(\frac{1}{2}-\frac{1}{2}\sqrt{5})$ 的幂. □

选读: 下面的例子给出有复数根和重根的递推关系的解.

例 4　复数根和重根

求满足关系 $a_n=-2a_{n-2}-a_{n-4}$ 且 $a_0=0,a_1=1,a_2=2,a_3=3$ 的 a_n 的公式.

做替换 $a_n=\alpha^n$, 我们得到 $\alpha^n=-2\alpha^{n-2}-\alpha^{n-4}$, 这产生特征方程 $\alpha^4+2\alpha^2+1=(\alpha^2+1)^2=0$. 这个方程的根是 $\alpha=+\mathrm{i}$ 和 $\alpha=-\mathrm{i}$(其中 $\mathrm{i}=\sqrt{-1}$), 而且每个根都是二重根. 回想一下, 当 α 是二重根时, 相应的递推关系的解是 $a_n=\alpha^n$, $a_n=n\alpha^n$. 所以一般解是:

$$a_n=A_1\mathrm{i}^n+A_2n\mathrm{i}^n+A_3(-\mathrm{i})^n+A_4n(-1)^n$$

初始条件产生方程:

$$\begin{aligned}
0=a_0&=A_1\mathrm{i}^0+A_20\mathrm{i}^0+A_3(-\mathrm{i})^0+A_40(-\mathrm{i})^0=A_1+0+A_3+0\\
1=a_1&=A_1\mathrm{i}^1+A_21\mathrm{i}^1+A_3(-\mathrm{i})^1+A_41(-\mathrm{i})^1=\mathrm{i}(A_1+A_2-A_3-A_4)\\
2=a_2&=A_1\mathrm{i}^2+A_22\mathrm{i}^2+A_3(-\mathrm{i})^2+A_42(-\mathrm{i})^2=-A_1-2A_2-A_3-2A_4\\
3=a_3&=A_1\mathrm{i}^3+A_23\mathrm{i}^3+A_3(-\mathrm{i})^3+A_43(-\mathrm{i})^3=\mathrm{i}(-A_1-3A_2+A_3+3A_4)
\end{aligned}$$

求解这些包含 4 个未知的 A_i 的 4 个联立方程, 我们得到:

$$A_1=-\frac{3}{2}\mathrm{i} \qquad A_2=-\frac{1}{2}+\mathrm{i} \qquad A_3=\frac{3}{2}\mathrm{i} \qquad A_4=-\frac{1}{2}-\mathrm{i}$$

于是, 递推关系的解是:

$$a_n=-\frac{3}{2}\mathrm{i}^{n+1}+\left(-\frac{1}{2}+\mathrm{i}\right)n\mathrm{i}^n+\frac{3}{2}\mathrm{i}(-\mathrm{i})^n+\left(-\frac{1}{2}-\mathrm{i}\right)n(-\mathrm{i})^n$$ □

我们提醒读者, 对于 n 的特殊值, 例如 $n=12$, 在前面的两个例子中, 很容易从递推关系出发, 递推计算 a_3,a_4,a_5 直到 a_{12}.

练 习 7.3

1. 如果账户上投资的 500 美元每年赚 8%, 给出经过 n 年后账户上钱的数额公式.
2. 求解任意两个连续数字都不相同的 n 位三进制序列个数的递推关系.
3. 求解下面的递推关系:

   (a) $a_n = 3a_{n-1} + 4a_{n-2}, a_0 = a_1 = 1$; (b) $a_n = a_{n-2}, a_0 = a_1 = 1$;

   (c) $a_n = 2a_{n-1} - a_{n-2}, a_0 = a_1 = 2$;

   (d) $a_n = 3a_{n-1} - 3a_{n-2} + a_{n-3}, a_0 = a_1 = 1, a_2 = 2$.
4. 确定例 3 中的常数 A_1 和 A_2. 首先证明初始条件 $a_1 = 1, a_2 = 2$ 等价于 $a_0 = 1, a_1 = 1$.
5. 求解下面的旗的排列方法数的递推关系: 在一根 n 英尺长的旗杆上使用 3 种类型的旗, 即高 2 英尺的红旗、高 1 英尺的黄旗和高 1 英尺的蓝旗.
6. 求解使用红、白和蓝 3 种薄片堆成 n 个薄片的一堆, 使得两张红色薄片不在一起的方法数的递推关系.
7. 求解 p_n 的递推关系, 其中 p_n 是股票市场的指数, 按着交易规则这一年 (从上一年开始) 的变动等于上一年变动的两倍. 假设 $p_0 = 1, p_1 = 4$.
8. 证明: (1) 的解的任意线性组合本身还是 (1) 的解.
9. 证明: 如果特征方程 (4) 有 3 重根 α_*, 那么对于 $j = 0, 1, 2, n^j\alpha_*^n$ 是 (1) 的解.
10. 证明: 如果 F_n 是从 $F_0 = F_1 = 1$ 开始的斐波那契序列中的第 n 个斐波那契数, 那么有:

$$\lim_{n\to\infty}\frac{F_{n+1}}{F_n} = \frac{1+\sqrt{5}}{2}$$

11. 如果递推关系 $a_n = c_1a_{n-1} + c_2a_{n-2}$ 有一般解 $a_n = A_13^n + A_26^n$, 求 c_1 和 c_2.

7.4 非齐次递推关系的解

递推关系称为是**齐次的**(homogeneous), 如果这个关系的所有项都包含某个 a_k. 在 7.3 节中, 我们给出了求解任意齐次线性递推关系 $a_n = c_1a_{n-1} + c_2a_{n-2} + \cdots + c_ra_{n-r}$ 的方法. 如果递推关系还包含一个常数项或一个 n 的函数项, 比如关系

$$a_n = ca_{n-1} + f(n) \tag{1}$$

其中 c 是常数, $f(n)$ 是 n 的函数, 那么这个递推关系就称为是 **非齐次的**(inhomogeneous).

本节讨论如何求解形如 (1) 的非齐次递推关系的方法. 求解这些递推头系的关键想法是, 非齐次递推关系的一般解是由相关的 [消去 $f(n)$ 项得到的] 齐次递推关系的一般解再加上这个非齐次递推关系的一个特殊解组成的.

对于 (1), 齐次递推关系是 $a_n = ca_{n-1}$, 这个齐次递推关系的一般解是 $a_n = Ac^n$. 假设 a_n^* 是 (1) 的某个特殊解, 即 $a_n^* = ca_{n-1}^* + f(n)$. 那么, $a_n = Ac^n + a_n^*$ 满足 (1):

$$\begin{aligned} a_n = Ac^n + a_n^* &= Ac^n + [ca_{n-1}^* + f(n)] \\ &= c(Ac^{n-1} + a_{n-1}^*) + f(n) = ca_{n-1} + f(n) \end{aligned}$$

在这个一般解中选择常数 A 以满足初始条件, 正如 7.3 节所做的那样 (只有求得 a_n* 后, 才能确定 A).

(1) 有一个特殊情况, 可以把这个特殊情况处理成为前几章中的枚举问题. 如果 $c=1$, 那么 (1) 变成:

$$a_n = a_{n-1} + f(n) \tag{2}$$

可以通过迭代 (2), 得到:

$$\begin{aligned} a_1 &= a_0 + f(1) \\ a_2 &= a_1 + f(2) = [a_0 + f(1)] + f(2) \\ a_3 &= a_2 + f(3) = [a_0 + f(1) + f(2)] + f(3) \\ &\ \vdots \\ a_n &= a_0 + f(1) + f(2) + f(3) + \cdots + f(n) = a_0 + \sum_{k=1}^{n} f(k) \end{aligned}$$

所以, a_n 是各 $f(k)$ 的和加上 a_0. 在 5.5 节和 5.6 节中, 我们给出对 n 的函数求和的方法. 其中的所有方法都可以用来求解 (2).

例 1 求和递推

在 7.1 节的例 3 中, 我们得到了 n 条两两相交的直线所创建的区域数 a_n 的递推关系 $a_n = a_{n-1} + n$, 其初始条件是 a_1=2.

可以把初始条件 a_1=2 替换成初始条件 a_0=1 (没有直线时, 平面是一个大区域). 根据前面的讨论, 我们看到 $a_n = 1 + (1 + 2 + 3 + \cdots + n)$. 这个求和的表达式可以写成下面的形式:

$$\binom{1}{1} + \binom{2}{1} + \binom{3}{1} + \cdots + \binom{n}{1} \tag{3}$$

根据 5.5 节的等式 (7), 这个和等于 $\binom{n+1}{2} = \frac{1}{2}n(n+1)$. 于是, $a_n = 1 + 1/2n(n+1)$. □

当 (1) 中的 $c \neq 1$ 时, 对于若干特定的函数 $f(n)$, 我们知道 (1) 的解, 这类似于 7.2 节给出的 "分治" 递推关系的情况. 在下面的表格中, 我们列出对应于若干最简单的 $f(n)$ 的解. 可以利用 7.5 节介绍的生成函数方法得到这些解.

$f(n)$	特殊解 $p(n)$
d, 一个常数	B
dn	$B_1n + B_0$
dn^2	$B_2n^2 + B_1n + B_0$
ed^n	Bd^n

表格中的 B 是待定的常数. 如果 $f(n)$ 是若干不同项的和, 那么可以首先分开求每一个 $f(n)$ 项的关系, 然后再把这些解加起来得到合成的 $f(n)$ 的特殊解.

存在 $f(n) = ed^n$ 的特殊解不能正常工作的情况. 这对应于关系 $a_n = da_{n-1} + ed^n$, 这里要注意的是, 齐次关系的解 $a_n = Ad^n$ 与 $f(n)$ 的形式相同. 这时, 你必须尝试着以 $a_n^* = Bnd^n$ 为特殊解.

例 2 求解汉诺塔问题

在 7.1 节的例 4 中, 我们得到进行 n 个圆环汉诺塔游戏时所需的移动次数 a_n 的递推关系 $a_n = 2a_{n-1} + 1$ 及 a_1=1. 现在, 我们来求解这个递推关系. 齐次方程 $a_n = 2a_{n-1}$ 的一般解是 $a_n = 2^n A$.

设 $a_n^* = B$ 来求这个非齐次关系的一个特殊解 [这是前面表格给出的 $f(n)$ 为常数时的特殊解形式]. 在这个关系中做替换, 我们得到

$$B = a_n^* = 2a_{n-1}^* + 1 = 2B + 1 \quad \text{或} \quad B = -1$$

所以 $a_n^* = -1$ 是特殊解, 因而一般非齐次解是 $a_n = 2^n A + a_n^* = 2^n A - 1$. 现在, 可以通过初始条件确定 $A : 1 = a_1 = 2^1 A - 1$ 或者 $1 = 2A - 1$. 因此 $A = 1$, 要求的解是 $a_n = 2^n - 1$. □

例 3 合成非齐次项

求解递推关系 $a_n = 3a_{n-1} - 4n + 3 \times 2^n$, 求得它的一般解. 同时求当 a_1=8 时的解.

齐次方程 $a_n = 3a_{n-1}$ 的一般解 $a_n = 3^n A$. 我们要为每一非齐次项所形成的关系求它们各自的特殊解. 首先, 求 $a_n = 3a_{n-1} - 4n$ 的特殊解, 采用形式 $a_n^* = B_1 n + B_0$, 得到

$$B_1 n + B_0 = a_n^* = 3a_{n-1}^* - 4n = 3[B_1(n-1) + B_0] - 4n. \tag{4}$$

现在, 令等式 (4) 两边的常数项与常数项相等, 且让 n 的系数与 n 的系数相等, 得:

$$\text{常数项}: \quad B_0 = -3B_1 + 3B_0 \tag{5}$$

$$n\ \text{项}: \quad B_1 n = 3B_1 n - 4n \quad \text{或} \quad B_1 = 3B_1 - 4 \tag{6}$$

通过等式 (6) 求解 B_1, 得 B_1=2. 把 B_1 代入到等式 (5), 我们得到 B_0=3. 所以, $a_n^* = 2n + 3$ 是 $a_n = 3a_{n-1} - 4n$ 的一个特殊解.

其次, 求 $a_n = 3a_{n-1} + 3 \times 2^n$ 的特殊解, 我们采用形式 $a_n^+ = 2^n B$, 得到:

$$2^n B = a_n^+ = 3a_{n-1}^+ + 3 \times 2^n = 3(2^{n-1}B) + 3 \times 2^n \tag{7}$$

等式 (7) 的两边同时除以 2^{n-1}, 我们得到 $2B = 3B + 6$, 于是 $B = -6$. 所以 $a_n^+ = -6 \times 2^n$ 是 $a_n = 3a_{n-1} + 3 \times 2^n$ 的一个特殊解. 把这两个特殊解与齐次关系

的一般解结合起来, 我们得到非齐次关系的一般解:

$$a_n = 3^n A + 2n + 3 - 6 \times 2^n$$

当 a_1=8 时, 能够如下确定 A:

$$8 = a_1 = 3^1 A + 2(1) + 3 - 6 \times 2^1 = 3A - 7$$

因此, $A = 5$, 而解是 $5 \times 3^n + 2n + 3 - 6 \times 2^n$. □

练 习 7.4

1. 求解下面的递推关系:

(a) $a_n = a_{n-1} + 3(n-1), a_0 = 1$; (b) $a_n = a_{n-1} + n(n-1), a_0 = 3$;

(c) $a_n = a_{n-1} + 3n^2, a_0 = 10$.

2. 求解平面上 n 条直线两两相交于不同点所形成的无限区域数的递推关系.

3. 求解在 $n \times n$ 棋盘上可以画出的任意大小方形子棋盘数的递推关系.

4. 在三维空间内有 m 个两两相交的平面, 如果任意 4 个平面不相交于同一个点, 且任意两个平面在第 3 个平面内没有平行相交直线, 求这样的方式所形成的不同区域个数的递推关系. [提示: 把问题简化成为二维平面问题 (例 1).]

5. 凸 n 边形的所有对角线把这个多边形分割成多个区域, 假设没有 3 条对角线相交于同一点, 求并解这些对角线所形成的区域个数的递推关系.

6. 如果连续两年产量的平均值 $1/2(a_n + a_{n-1})$ 是 $2n + 5$ 且 a_0=3, 求 a_n.

7. 从 7.1 节例 5 的部分 (b) 开始, 求解递推关系 $a_n = 1.04a_{n-1} + 100, a_0 = 0$.

8. 假设账户一年赚 5%. 最初在账户中有 1 000 美元, 而且在第 k 年取出 $10k$ 美元. 在 n 年末, 这个账户中有多少钱, 如果:

(a) 每年的取款是在年末进行的? (b) 每年的取款是在年初进行的?

9. 求解下面的递推关系:

(a) $a_n = 3a_{n-1} - 2, a_0 = 0$; (b) $a_n = 2a_{n-1} + (-1)^n, a_0 = 2$;

(c) $a_n = 2a_{n-1} + n, a_0 = 1$; (d) $a_n = 2a_{n-1} + 2n^2, a_0 = 3$.

10. 求解递推关系 $a_n = 3a_{n-1} + n^2 - 3$ 且 $a_0 = 1$.

11. 求解递推关系 $a_n = 3a_{n-1} - 2a_{n-2} + 3$ 且 $a_0 = a_1 = 1$.

12. 求解 1 不出现在任意 2 的右侧的 n 位三进制序列个数的递推关系.

13. 求解公司盈利的递推关系, 如果从前一年开始, 在第 k 年盈利增长的增长率是 10×2^k, 其中 $a_0 = 20, a_1 = 1\,020$.

14. 证明: 任意非齐次线性递推关系的一般解是相关的齐次关系的一般解加上非齐次关系的一个特殊解.

15. 证明: 本节表格中给出的 (1) 的特殊解形式对下面各函数是正确的:

(a) $f(n) = d$; (b) $f(n) = dn$; (c) $f(n) = dn^2$; (d) $f(n) = d^n$.

16. 证明: 如果 (1) 中的 $f(n)$ 是若干不同项的和, 那么可以通过把各个项的特殊解加起来得到这个函数 $f(n)$ 的特殊解.

17. 求 $a_n - 5a_{n-1} + 6a_{n-2} = 2 + 3n$ 的一般解.

18. 设递推关系 $a_n - c_1a_{n-1} + c_2a_{n-2} = c_3n + c_4$ 有一般解 $a_n = A_12^n + A_25^n + 3n - 5$, 求 c_1, c_2, c_3, c_4.

19. 求解下面的递推关系, 此时 a_0=1:

(a) $a_n^2 = 2a_{n-1}^2 + 1$ [提示: 设 $b_n = a_n^2$].

(b) $a_n = -na_{n-1} + n!$ [提示: 如问题 (a) 那样, 定义适当的 b_n].

7.5 使用生成函数对递推关系求解

大多数关于 a_n 的递推关系都可以转换成一个关于生成函数 $g(x) = a_0 + a_1x + \cdots + a_nx^n + \cdots$ 的方程. 通常可以用代数方法对这个相关 $g(x)$ 的函数方程求解, 把 $g(x)$ 的最终表达式以幂级数展开, 得到作为 x^n 的系数的 a_n. 这些函数方程的某些代数处理对读者来说可能是新鲜的.

我们把 $g(x)$ 当作如 y 一样的单一变量处理, 而把其他 x 的函数处理成常量. 例如, 我们可以把函数方程 $g(x) = x^2g(x) - 2x$ 重写成 $g(x)(1 - x^2) = -2x$, 因此 $g(x) = -2x(1 - x^2)^{-1}$, 从而解得这个函数方程. 类似地, 函数方程

$$(1 - x^2)[g(x)]^2 - 4xg(x) + 4x^2 = 0 \tag{1}$$

可以通过运用形如 $ay^2 + by + c = 0$ 这样的方程的二次公式来求解. 其中, $a = (1 - x^2), b = -4x, c = 4x^2$. 直观上, 对于 x 的每一个特殊值, $g(x)$ 都是 (1) 的解. 因此, 根据二次公式, (1) 的解是

$$g(x) = \frac{1}{2(1 - x^2)}[4x \pm \sqrt{16x^2 - 16x^2(1 - x^2)}] = \frac{1}{2(1 - x^2)}(4x \pm 4x^2)$$

所以 $g(x) = 2(x + x^2)/(1 - x^2)$, 或 $2(x - x^2)/(1 - x^2)$. 如果存在两个或更多的可能解, 那么只有一个是通常意义下的 a_n 的生成函数 (例如, 对于初始条件 a_0, 有一个有正确值的幂级数展开).

现在, 我们通过例子展示如何把在 7.1 节得到的各个递推关系转化成相关生成函数的函数方程.

例 1 求和递推

寻找 $g(x) = a_0 + a_1x + \cdots + a_nx^n + \cdots$ 的一个函数方程, 其中 a_n 满足 7.1 节例 3 中得到的递推关系 $a_n = a_{n-1} + n$, 且 $n \geqslant 1$. 初始条件是 a_0=1. 求解这个函数方程并展开 $g(x)$ 来得到 a_n.

除 a_0 外, 对 $g(x)$ 中的每一项使用这个递推关系, 有 $a_nx^n = a_{n-1}x^n + nx^n$, $n \geqslant 1$. 把这些项加起来, 我们可以写出:

$$g(x) - a_0 = \sum_{n=1}^{\infty} a_nx^n = \sum_{n=1}^{\infty}(a_{n-1}x^n + nx^n) \tag{2}$$

$$= x\sum_{n=1}^{\infty} a_{n-1}x^{n-1} + \sum_{n=1}^{\infty} nx^n \tag{3}$$

$$= x\sum_{m=0}^{\infty} a_m x^m + \sum_{n=0}^{\infty} \binom{n}{1} x^n \tag{4}$$

$$= xg(x) + \frac{x}{(1-x)^2} \tag{5}$$

把行 (2) 中的两个 x^n 的和分解成各项的和, 再把 $a_{n-1}x^n$ 改写成 $xa_{n-1}x^{n-1}$(为了使 x 的幂与 a_{n-1} 的下标一致) 就可得到行 (3). 从行 (3) 开始, 把第一个和用 $m=n-1$ 来改变下标, 并在第二个和中加入 0 项 $0x^0$, 再改写 n 为 $C(n,1)$. 第一个级数是生成函数 $g(x)$ 乘以 x. 第二个级数是利用 6.5 节给出的构造方法而得到的生成函数. 行 (5) 与 $g(x)-a_0$ 相等 [行 (2) 的左边], 并设 a_0=1, 我们得到所要求的 $g(x)$ 的函数方程:

$$g(x) - 1 = xg(x) + \frac{x}{(1-x)^2} \tag{6}$$

为了解得 $g(x)$, 我们把 (6) 改写成:

$$g(x) - xg(x) = 1 + \frac{x}{(1-x)^2} \quad 或 \quad g(x)(1-x) = 1 + \frac{x}{(1-x)^2}$$

因此有

$$g(x) = \frac{1}{(1-x)} + \frac{x}{(1-x)^3}$$

x^n 在 $(1-x)^{-1}$ 中的系数是 1, 在 $x(1-x)^{-3}$ 中的系数是 $C((n-1)+3-1, n-1) = C(n+1, n-1) = C(n+1, 2)$. 于是

$$g(x) = \frac{1}{(1-x)} + \frac{x}{(1-x)^3} = \sum_{n=0}^{\infty} x^n + \sum_{n=0}^{\infty} \binom{n+1}{2} x^n = \sum_{n=0}^{\infty} \left[1 + \binom{n+1}{2}\right] x^n$$

所以, $a_n = 1 + C(n+1, 2)$, 与 7.4 节例 1 中给出这个递推关系的答案相同. □

例 2　斐波那契关系

使用生成函数解 7.1 节例 2 得到的递推关系 $a_n = a_{n-1} + a_{n-2}$, 其中 $a_1 = 1, a_2$=2.

初始条件 $a_1 = 1, a_2 = 2$ 等价于 $a_0 = 1, a_1 = 1$. 如前面例子那样, 使用相同的幂级数和方法, 我们得到

$$\begin{aligned} g(x) - a_0 - a_1 x = \sum_{n=2}^{\infty} a_n x^n &= \sum_{n=2}^{\infty} (a_{n-1}x^n + a_{n-2}x^n) \\ &= x\sum_{n=2}^{\infty} a_{n-1}x^{n-1} + x^2\sum_{n=2}^{\infty} a_{n-2}x^{n-2} = x\sum_{m=1}^{\infty} a_m x^m + x^2 \sum_{k=0}^{\infty} a_k x^k \\ &= x[g(x) - a_0] + x^2 g(x) \end{aligned}$$

设 $a_0=1, a_1=1$, 我们得到函数方程

$$g(x)-1-x=x[g(x)-1]+x^2g(x) \quad 或 \quad g(x)-xg(x)-x^2g(x)-1=0$$

所以有

$$g(x)(1-x-x^2)=1 \quad 或 \quad g(x)=1/(1-x-x^2)$$

观察发现, 上面函数方程中的分母与 7.3 节例 3 中给出的递推关系的特征方程关系密切. 一般地, $g(x)$ 有分母 $1+c_1x+c_2x^2+\cdots+c_rx^r$ 当且仅当 $x^r+c_1x^{r-1}+c_2x^{r-2}+\cdots+c_r$ 是相应递推关系的特征方程, 所以 $1-\alpha x$ 是 $g(x)$ 的分母的一个因子当且仅当 α 是这个特征方程的根.

分子与初始条件有关.

根据二次公式, 我们可以因此分解得:

$$1-x-x^2=\left[1-\frac{1}{2}(1+\sqrt{5})x\right]\left[1-\frac{1}{2}(1-\sqrt{5})x\right]$$

为了简便起见, 定义 $\alpha_1=\frac{1}{2}(1+\sqrt{5}), \alpha_2=\frac{1}{2}(1-\sqrt{5})$. 于是有

$$g(x)=\frac{1}{(1-\alpha_1x)(1-\alpha_2x)}=\frac{\alpha_1/\sqrt{5}}{1-\alpha_1x}-\frac{\alpha_2/\sqrt{5}}{1-\alpha_2x} \tag{7}$$

可以利用部分分式方法把 $g(x)$ 分解成 (7) 中的两个分式形式, 这是把两个分式合并成一个公分式的过程的逆过程. 在 (7) 右边的第一个分式中设 $y=\alpha_1x$, 我们有:

$$\frac{\alpha_1/\sqrt{5}}{1-\alpha_1x}=\frac{\alpha_1}{\sqrt{5}}\left(\frac{1}{1-y}\right)=\frac{\alpha_1}{\sqrt{5}}\sum_{n=0}^{\infty}y^n=\frac{\alpha_1}{\sqrt{5}}\sum_{n=0}^{\infty}\alpha_1^nx^n$$

$y=\alpha_2x$ 有相同的展开式类型. 于是 $g(x)$ 的幂级数展开式中的 x^n 的系数 a_n 是

$$a_n=\frac{1}{\sqrt{5}}\alpha_1^{n+1}-\frac{1}{\sqrt{5}}\alpha_2^{n+1}=\frac{1}{\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right)^{n+1}-\frac{1}{\sqrt{5}}\left(\frac{1-\sqrt{5}}{2}\right)^{n+1} \qquad \square$$

观察发现, 使用斐波那契关系递推地计算 $a_2, a_3, \cdots$, 一直到 a_{10}, 比在前面的 a_n 的公式中设 $n=10$ 来得到 a_{10} 要容易得多.

例 3 无重复选取

设 $g_n(x)$ 是这样的一组生成函数 $g_n(x)=a_{n,0}+a_{n,1}x+\cdots+a_{n,k}x^k+\cdots+a_{n,n}x^n$, 它满足 7.1 节例 8 得到的 n 集合的 k 元子集数量 $a_{n,k}$ 的关系 $a_{n,k}=a_{n-1,k}+a_{n-1,k-1}, a_{n,0}=a_{n,n}=1$ $(k>n, a_{n,k}=0)$. 求关于 $g_n(x)$ 的函数关系, 并求解这个函数关系而得到 $a_{n,k}$ 的公式. 使用幂级数和方法, 我们得到

$$g_n(x)-1=\sum_{k=1}^{n}a_{n,k}x^k=\sum_{k=1}^{n}(a_{n-1,k}x^k+a_{n-1,k-1}x^k)$$

$$= \sum_{k=1}^{n} a_{n-1,k}x^k + x\sum_{h=0}^{n-1} a_{n-1,h}x^h = g_{n-1}(x) - 1 + xg_{n-1}(x)$$

因此

$$g_n(x) = g_{n-1}(x) + xg_{n-1}(x) = (1+x)g_{n-1}(x)$$

我们求得了最终的关系 $g_n(x) = (1+x)g_{n-1}(x)$, 这一形式类似于 $a_n = ca_{n-1}$. 因为 $g_0(x) = a_{0,0} = 1$, 所以解是

$$g_n(x) = (1+x)^n g_0(x) = (1+x)^n$$

根据二项式定理, 我们有 $a_{n,k} = C(n,k)$. □

选读: 本节剩余的内容涉及更复杂的计算. 读者可以跳过这些内容. 首先考虑一个非线性递推关系, 并使用生成函数对其求解.

例 4 放置括号

求解 7.1 节例 10 得到的递推关系 $a_n = a_1a_{n-1} + a_2a_{n-2} + \cdots + a_ia_{n-i} + \cdots + a_{n-1}a_1$, 它是 n 个数相乘时放置括号的方法数的递推关系.

观察发现, 如果 $g(x) = a_0 + a_1x + \cdots + a_nx^n + \cdots$, 那么上面方程的右边就是对于 $n \geqslant 2$, x^n 在积 $g(x)g(x) = (0 + a_1x + \cdots + a_nx^n + \cdots)^2$ 中的系数. 使用幂级数和方法, 我们有 (回想一下 $a_0 = 0, a_1 = 1$):

$$g(x) - 1x = \sum_{n=2}^{\infty} a_nx^n = \sum_{n=2}^{\infty}(a_1a_{n-1} + a_2a_{n-2} + \cdots + a_{n-1}a_1)x^n = [g(x)]^2$$

如本节开始所述, 解 $g(x)$ 的二次方程, 我们得到 $g(x) = \frac{1}{2}(1 \pm \sqrt{1-4x})$. 设 $a_0 = 0$[即, $g(0) = 0$], 我们想要的解是 $g(x) = \frac{1}{2} - \frac{1}{2}\sqrt{1-4x}$.

$g(x)$ 需要一个新型的生成函数展开式, 我们称这个新型的展开式为*广义二项式定理*(generalized binomial theorem). 对于任意实数 q, 幂级数展开式 $(1+y)^q = \binom{q}{0} + \binom{q}{1}y + \binom{q}{2}y^2 + \cdots + \binom{q}{n}y^n + \cdots$ 中 y_n 的系数 $\binom{q}{n}$ 定义为:

$$\binom{q}{n} = \frac{q(q-1)(q-2) \times \cdots \times [q-(n-1)]}{n!} \tag{8}$$

[上面这个一般系数 $\binom{q}{n}$ 的公式来自于 $(1+y)^q$ 的泰勒级数, 参见其他标准的微积分教程.]

使用 (8), x^n 在 $\sqrt{1-4x}$ 中的系数是 [把 $\sqrt{1-4x}$ 想成是 $(1+y)^{1/2}$, 其中 $y = -4x$]:

$$\begin{aligned}\binom{\frac{1}{2}}{n}(-4)^n &= \frac{\frac{1}{2}(-\frac{1}{2})(-\frac{3}{2}) \times \cdots \times [-\frac{1}{2}(2n-3)]}{n!}(-4)^n \\ &= \frac{-1 \times 1 \times 3 \times 5 \times \cdots \times (2n-3)}{n!}2^n = -\frac{2}{n}\binom{2n-2}{n-1}\end{aligned} \tag{9}$$

把上式第二步的分子中的数与 2 的适当幂相乘就可以得到 (9) 的最终结果 (细节留作练习 6). 把等式 (9) 中的最后表达式乘以 $-\frac{1}{2}$, 我们得到 x^n 在 $-\frac{1}{2}\sqrt{1-4x}$ 中的系数.

$$a_n = \frac{1}{n}\binom{2n-2}{n-1}; \quad n \geqslant 1 \qquad \square$$

表达式 $\frac{1}{2}C(2n-2, n-1)$ 出现于各组合环境中, 我们称之为卡塔兰数(Catalan number). 需要注意的是, 当数相乘时, 无论括号放置在哪里, 都不会影响结果, 但如果我们正在做的是矩阵乘积时, 括号的放置对结果产生重要的影响. 下面考虑联立递推关系的生成函数.

例 5　联立递推关系

使用生成函数解 7.1 节例 11 得到的联立递推关系.

$$a_n = a_{n-1} + b_{n-1} + c_{n-1}, \quad b_n = 3^{n-1} - c_{c-1}, \quad c_n = 3^{n-1} - b_{n-1}, \quad a_1 = b_1 = c_1 = 1$$

设 $A(x), B(x), C(x)$ 分别是 a_n, b_n, c_n 的生成函数. 我们使用幂级数和的方法得到

$$\begin{aligned}
A(x) - a_0 = \sum_{n=1}^{\infty} a_n x^n &= \sum_{n=1}^{\infty} a_{n-1}x^n + \sum_{n=1}^{\infty} b_{n-1}x^n + \sum_{n=1}^{\infty} c_{n-1}x^n \\
&= x\sum_{m=0}^{\infty} a_m x^m + x\sum_{m=0}^{\infty} b_m x^m + x\sum_{m=0}^{\infty} c_m x^m \\
&= xA(x) + xB(x) + xC(x) \\
B(x) - b_0 = \sum_{n=1}^{\infty} b_n x^n &= \sum_{n=1}^{\infty} 3^{n-1}x^n - \sum_{n=1}^{\infty} c_{n-1}x^n \\
&= x(1-3x)^{-1} - xC(x) \\
C(x) - c_0 = \sum_{n=1}^{\infty} c_n x^n &= \sum_{n=1}^{\infty} 3^{n-1}x^n - \sum_{n=1}^{\infty} b_{n-1}x^n \\
&= x(1-3x)^{-1} - xB(x)
\end{aligned}$$

用 a_0, b_0, c_0 描述初始条件总是最为合适. 给定 $a_1 = b_1 = c_1 = 1$, 解 a_0, b_0, c_0 的 3 个递推关系, 我们得到 $1 = b_1 = 3^0 - c_0$, 即 c_0=0. 类似地得到 $b_0 = 0, a_0$=1. 于是我们的函数方程是:

$$A(x) - 1 = xA(x) + xB(x) + xC(x)$$

或

$$A(x) = \frac{1}{1-x}[xB(x) + xC(x) + 1] \tag{10}$$

及

$$B(x) = \frac{x}{1-3x} - xC(x) \tag{11}$$

$$C(x) = \frac{x}{1-3x} - xB(x) \tag{12}$$

可以对关于 $B(x)$ 和 $C(x)$ 的 (11) 和 (12) 联立求解. 把 (12) 乘以 x, 并使用 (11) 中的 $xC(x)$ 的表达式, 我们有

$$\begin{aligned}
B(x) &= \frac{x}{1-3x} - xC(x) = \frac{x}{1-3x} - x\left[\frac{x}{1-3x} - xB(x)\right] \\
&\Rightarrow B(x)(1-x^2) = \frac{x-x^2}{1-3x} \\
&\Rightarrow B(x) = \frac{(1-x)x}{(1-x^2)(1-3x)} = \frac{x}{(1+x)(1-3x)} = \frac{\frac{1}{4}}{1-3x} - \frac{\frac{1}{4}}{1+x}
\end{aligned}$$

上面的计算的最后一步还是一个部分分式分解. x^n 在 $\frac{1}{4}(1-3x)^{-1}$ 中的系数是 $\frac{1}{4}\times 3^n$, 而在 $-\frac{1}{4}(1+x)^{-1}$ 的系数是 $-\frac{1}{4}(-1)^n$. x^n 在 $B(x)$ 中的系数 b_n 是 $\frac{1}{4}[3^n - (-1)^n]$. 方程 (11) 和 (12) 对 $B(x)$ 和 $C(x)$ 是对称的, 所以 $C(x) = B(x)$ 且 $c_n = b_n = \frac{1}{4}[3^n - (-1)^n]$.

接下来, 对 (10) 中的 $A(x)$ 求解:

$$\begin{aligned}
A(x) &= \frac{1}{1-x}[xB(x) + xC(x) + 1] = \frac{2x}{1-x}B(x) + \frac{1}{1-x} \quad [\text{因为 } B(x) = C(x)] \\
&= \frac{2x}{1-x}\left(\frac{\frac{1}{4}}{1-3x} - \frac{\frac{1}{4}}{1+x}\right) + \frac{1}{1-x} = \frac{\frac{1}{2}x}{(1-x)(1-3x)} - \frac{\frac{1}{2}x}{1-x^2} + \frac{1}{1-x} \\
&= \left(\frac{\frac{1}{4}}{1-3x} - \frac{\frac{1}{4}}{1-x}\right) - \frac{\frac{1}{2}x}{1-x^2} + \frac{1}{1-x}
\end{aligned}$$

x^n 在 $\frac{1}{4}(1-3x)^{-1}$ 中的系数是 $\frac{1}{4}\times 3^n$, 在 $-\frac{1}{4}(1-x)^{-1}$ 中的系数是 $-\frac{1}{4}$, 当 n 是奇数时, x^n 在 $-\frac{x}{2}(1-x^2)^{-1}$ 中的系数是 $-\frac{1}{2}$, 当 n 是偶数时, 这个系数是 0, x^n 在 $(1-x)^{-1}$ 中的系数 1. 把这些项结合到一起, 我们得到: 当 n 是偶数时, $a_n = \frac{1}{4}(3^n+3)$; 当 n 是奇数时, $a_n = \frac{1}{4}(3^n+1)$. □

练　习　7.5

1. 寻找系数满足下面关系的生成函数的函数方程:
   (a) $a_n = a_{n-1} + 2, a_0 = 1$;
   (b) $a_n = 3a_{n-1} - 2a_{n-2} + 2, a_0 = a_1 = 1$;
   (c) $a_n = a_{n-1} + n(n-1), a_0 = 1$;
   (d) $a_n = 2a_{n-1} + 2^n, a_0 = 1$.
2. 使用生成函数对练习 1 中的递推关系求解.
3. 寻找系数满足下面关系的生成函数的函数方程:
   (a) $a_n = \sum_{i=0}^{n-1} a_i a_{n-1-i} (n \geqslant 1), a_0 = 1$;
   (b) $a_n = \sum_{i=2}^{n-2} a_i a_{n-i} (n \geqslant 3), a_0 = a_1 = a_2 = 1$;
   (c) $a_n = \sum_{i=1}^{n-1} 2^i a_{n-i} (n \geqslant 2), a_0 = a_1 = 1$.

4. 对于系数满足下面条件的生成函数序列 $F_n(x)=\sum\limits_{k=0}^{n}a_{n,k}x^k$, 寻找函数方程, 并对其求解.

(a) $a_{n,k}=a_{n,k-1}-2a_{n-1,k-1}, a_{n,0}=0$. **(b)** $a_{n,k}=2a_{n-1,k}-3a_{n,k-1}, a_{n,0}=1$.

5. 证明 7.4 节的表格中的非齐次递推关系的特殊解形式.

6. 证明 (9) 中的下面结果:

$$\frac{-1\times 1\times 3\times 5\times\cdots\times(2n-3)}{n!}2^n=-\frac{2}{n}\binom{2n-2}{n-1}$$

7. 寻找没有相交弦的凸 n 边形的三角形分割方法数的递推关系, 并利用生成函数求解这个递推关系. (提示: 利用与例 4 类似的推理.)

8. 对于 n 位三进制序列中模式 "012" 的第一次出现是在序列末尾的序列个数, 寻找递推关系和相关生成函数.

9. 对于有 n 个叶子的不同二叉树个数, 寻找递推关系和相关的生成函数.

10. 寻找 n 元集合的重复 k 子集个数 $a_{n,k}$ 的递推关系. 寻找 $F_n(x)=\sum\limits_{k=0}^{\infty}a_{n,k}x^k$ 的一个方程, 然后求解 $F_n(x)$ 和 $a_{n,k}$.

11. 设一次实验中, 每次试验成功的概率是 p. 设 $a_{n,k}$ 是在 n 次试验中出现 k 次成功的概率. 寻找 $a_{n,k}$ 的递推关系. 使用这一关系寻找 $F_n(x)=\sum\limits_{k=0}^{\infty}a_{n,k}x^k$ 的方程并对其求解.

12. **(a)** 寻找 n 个元素的 k 置换数 $a_{n,k}$ 的递推关系.

(b) 证明: $F_n(x)=\sum\limits_{k=0}^{\infty}a_{n,k}x^k$ 满足微分方程 [$F'_{n-1}(x)$ 表示 $F_{n-1}(x)$ 的导数]

$$F_n(x)=(1+x)F_{n-1}(x)+x^2F'_{n-1}(x)$$

(c) 求 $G_n(x)=\sum\limits_{k=0}^{n}a_{n,k}x^k/k!$ 的函数方程.

13. 寻找一个递推关系系统, 利用这个递推关系系统可以确定有奇数个 1 和奇数个 2 的 n 位四进制序列的个数. 求解这一递推关系系统.

14. 寻找一组联立递推关系, 用它们可以确定数字和是 3 的倍数的 n 位三进制序列的个数. 求解这一联立递推关系.

15. **(a)** 寻找 a_n 的递推关系, 其中 a_n 是 n 个不同对象到 n 个相同的盒子的分配方法数 (某些盒子可能是空的).

(b) 设$g(x)=\sum\limits_{n=0}^{\infty}a_nx^n/n!$, 其中 a_0=1. 证明: $g(x)$ 满足微分方程 $g'(x)=g(x)\mathrm{e}^x$. 对这一方程求解 $g(x)$.

16. **(a)** 定义 $S_{n,r}$ 为使得$\sum\limits_{r=0}^{n}S_{n,r}x^r=x(x-1)(x-2)\cdots$ 成立的数. 寻找 $S_{n,r}$ 的递推关系.

(b) 寻找 $F_n(x)=\sum\limits_{r=0}^{n}S_{n,r}x^r/r!$ 的一个微分方程.

7.6 小结及参考文献

在本章, 我们看到递推关系是解决计数问题的最简单的方法之一. 不像第 5 章要求的那样, 我们不需要对组合过程有充分的了解, 只需把给定的 n 对象问题用更

少对象的相同问题表示出来即可. 一旦求得一个递推关系, 那么从 a_1 开始 (一个对象时的解), 就可以连续地求 2 个对象、3 个对象、$\cdots$、直至 n 个对象的解. 或者我们可以尝试着用本章最后几节给出的技术来对这个递推关系求解.

数学著作中出现的第一个递推关系是斐波那契关系. 数学家 Leonardo di Pisa, 也即斐波那契, 在他 1220 年出版的著作《算盘书》中提出了关于兔子数量增长的计数问题 (7.1 节的练习 7). 经过 n 个月后的兔子数量 a_n 满足斐波那契关系 $a_n = a_{n-1} + a_{n-2}$. 正如在 6.6 节所提到的那样, 500 年后的 1730 年, DeMoivre 使用生成函数给出了 7.5 节例 2 中推导的这个关系的第一个解. 斐波那契关系和斐波那契数令人惊讶地到处存在. 例如, 当我们要搜索函数在某个区间的最小值时, 斐波那契数为分割这个区间 (在某种意义上) 提供一种最优方法. (参见 Kiefer[3].) Adler[1] 讨论了花周围的叶环中的斐波那契数.

递推关系的求解方法最初出现于差分方程以及相关联的微分方程理论的发展中. 关于差分方程的方法及其应用的较好描述, 参见 Sandefur[4].

1. I. Adler, "The consequence of constant pressure in phyllotaxis," *J. Theoretical Biology* 65(1977), 29–77.

2. T. Cormen, C. Leiserson, and R. Rivest, *An Introduction to Algorithms*, 2nd ed., MIT Press, Cambridge, MA, 2001.

3. J. Kiefer, "Sequential minimax search for a maximum," *Proceedings of American Math. Society* 4(1953), 502–506.

4. J. Sandefur, *Discrete Dynamical Modeling,* Oxford University Press, New York, 1993.

第8章 容斥原理

8.1 利用 Venn 图计数

本章研究一组用来解决一类计数问题的集合论公式, 这类计数问题包含若干或者同时都成立或者同时都不成立的相互影响的性质. 这样的一个例子是计数每种花色至少有一张牌的所有 5 张手牌的数量, 或等价地计数任何花色都非空的 5 张手牌的数量. 在解决这样的问题的过程中, 我们必须计数这些性质成立的各种组合中所要的结果子集. 我们使用 Venn 图描述这些不同的组合. 关于集合论和 Venn 图的讨论, 参阅附录 A.1.

我们从单一性质 Venn 图开始, 然后再进一步讨论有两三个性质的问题. 在图 8.1 中, 我们给出全集 $\mathscr{U}$ 内的一个集合 A. 其补集合 $\overline{A}$ 是由 $\mathscr{U}$ 的所有不在 A 中的元素组成的. 设 $N(S)$ 表示 S 中的元素个数. 我们定义 $N=N(\mathscr{U})$. 那么 $N(A)=N-N(\overline{A})$, 或者 $N(\overline{A})=N-N(A)$. 这一公式类似于概率中的情况, 某一事件 E 的概率等于 1 减去其补事件 $\overline{E}$ 的概率.

例如, 假设一门数学课程有 100 名学生的一个全集学习, 其中有 30 名学生不主修这门课程, 即 $N(\overline{A})$=30, 其中 A 是主修这门课程的学生. 于是, 主修这门数学课程的人数 $N(A)$ 是 $N(A)=N-N(\overline{A})=100-30=70$.

接下来, 考虑涉及两个集合的问题. 设全集 $\mathscr{U}$ 是一所学校里的所有学生, F 是选修法语的学生集合, L 是选修拉丁语的学生集合. 参见图 8.2. 我们想用 $N, N(F)$, $N(L)$ 和 $N(F\cap L)$ 来表示选修法语或者拉丁语的学生数 $N(F\cup L)$, 以及二者都不选修的学生数 $N(\overline{F}\cap\overline{L})$ 的公式. 注意, $N(F\cup L)$ 不是简单的 $N(F)+N(L)$, 因为 $N(F)+N(L)$ 把同时选修两种语言的每名学生计数了两次. 因此, 必须知道有多少学生选修了两种语言, 即 $N(F\cap L)$. 从 $N(F)+N(L)$ 中减去 $N(F\cap L)$ 来修正把选修两种语言的学生计数两次的错误. 即,

$$N(F\cup L)=N(F)+N(L)-N(F\cap L) \tag{1}$$

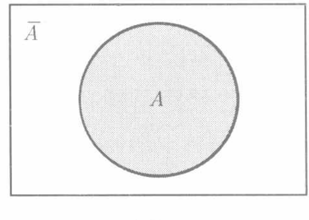


图 8.1

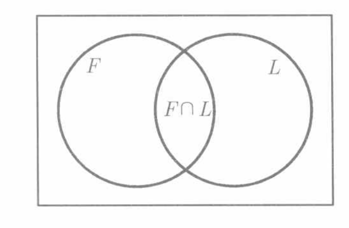


图 8.2

根据德摩根律 (附录 A.1 的等式 BA3),$\overline{F} \cap \overline{L} = \overline{F \cup L}$, 所以有

$$N(\overline{F} \cap \overline{L}) = N(\overline{F \cup L}) = N - N(F \cup L)$$

把这个等式与 (1) 相结合, 我们有

$$N(\overline{F} \cap \overline{L}) = N - N(F \cup L) = N - N(F) - N(L) + N(F \cap L) \tag{2}$$

例 1 不选修任何语言的学生

如果一所学校有 100 名学生, 而 50 名学生选修了法语, 40 名学生选修了拉丁语, 20 名学生选修了两种语言, 有多少名学生不选修任何语言?

在这一问题中, $N = 100, N(F) = 50, N(L) = 40, N(F \cap L) = 20$. 我们需要确定 $N(\overline{F} \cap \overline{L})$. 根据等式 (2), 我们有

$$N(\overline{F} \cap \overline{L}) = N - N(F) - N(L) + N(F \cap L) = 100 - 50 - 40 + 20 = 30 \qquad \square$$

下一个例子把这一集合论公式运用在没有明显集合结构的计数问题上.

例 2 限定排列

数字 $0, 1, 2, \cdots, 9$ 的排列中第一位数字大于 1 而最后一位数字小于 8 的排列有多少?

在利用公式 (2) 解决这一问题之前, 考虑一种更直接的方法, 并了解一下, 这种方法为什么不行. 对于这样的排列来说, 第一位数字有 8 种选择. 最后一位数字可选择的数依赖于第一位数字是否选择 8 或 9. 如果第一位数字选择 8 或 9, 最后一位数字有 8 种选择, 而第一位数字即不选择 8 也不选择 9, 那么最后一位数字有 7 种选择. 由于这样的困难, 我们将利用公式 (2) 来解决这一问题.

设 $\mathscr{U}$ 是 $0, 1, 2, \cdots, 9$ 的所有排列集合. 设 F 是第一位数字是 0 或 1 的所有排列的集合, L 是最后一位数字是 8 或者 9 的所有排列的集合. 那么第一位数字大于 1 而最后一位数字小于 8 的排列数是 $N(\overline{F} \cap \overline{L})$.

我们有 $N = 10!, N(F) = 2 \times 9!$ (第一个数字有 2 种选择, 其后是剩余 9 个数字的任意排列), 类似地, $N(L) = 2 \times 9!$, $N(F \cap L) = 2 \times 2 \times 8!$. 于是, 根据 (2), 有

$$N(\overline{F} \cap \overline{L}) = 10! - (2 \times 9!) - (2 \times 9!) + (2 \times 2 \times 8!)$$

$\square$

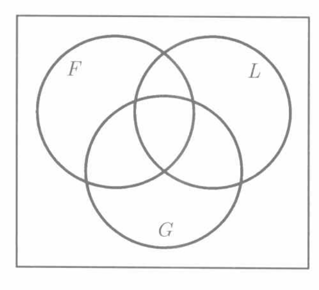


图 8.3

下面考虑涉及 3 个集合的问题. 我们额外附加选修德语的学生集合 G 来扩展图 8.2, 如图 8.3 所示. 我们想求 $N(\overline{F} \cap \overline{L} \cap \overline{G})$ 的公式. 第一个猜测是:

$$N(\overline{F} \cap \overline{L} \cap \overline{G}) \stackrel{?}{=} N - N(F) - N(L) - N(G)$$

正如在图 8.2 中那样, 这个公式把 F, L, G 中的两个集合中的学生计数了 2 次 (即减去 2 次). 我们加入同时选修这两种语言的学生数量来修正这一公式. 因此我们提出下面的公式:

$$N(\overline{F}\cap\overline{L}\cap\overline{G})\overset{?}{=}N-N(F)-N(L)-N(G)$$
$$+N(F\cap L)+N(L\cap G)+N(F\cap G) \tag{3}$$

图 8.4 给出了一名学生在上面公式的各个部分中被加上和减去的次数. 不选修任何语言的学生在项 N 中计数一次, 这一类学生正是我们要计数的学生. 具有挑战性的问题是要确保所有其他学生的净计数是 0 次. 选修一种语言的学生在 N 中计数一次, 而且在项 $-[N(F)+N(L)+N(G)]$ 中被减去一次, 所以净计数是 0 次. 选修两种语言的学生在 N 中计数一次, 而又在项 $-[N(F)+N(L)+N(G)]$ 中被减去两次 (因为他们正好在 3 个集合中的两个集合中), 随后又在项 $+[N(F\cap L)+N(L\cap G)+N(F\cap G)]$ 被加上一次 (因为他们正好在两两相交的集合中的一对集合中), 所以净计数 0 次. 最后, 考虑选修所有 3 种语言的学生. 他们由 N 计数一次, 然后, 由 3 个集合的和减去 3 次 (因为他们在所有 3 个集合中), 又由 3 对两两相交的集合加上 3 次 (因为他们在所有 3 对集合中). 他们的净计数是 $1-3+3=1$. 所以, 必须修正公式 (3), 即减去 $N(F\cap L\cap G)$, 使得选修 3 种语言的学生的净计数成为 0:

$$N(\overline{F}\cap\overline{L}\cap\overline{G})=N-[N(F)+N(L)+N(G)]+[N(F\cap L)+N(L\cap G)$$
$$+N(F\cap G)]-N(F\cap L\cap G) \tag{4}$$

对于一般的集合 A_1, A_2, A_3, 我们重写 (4) 为:

$$N(\overline{A}_1\cap\overline{A}_2\cap\overline{A}_3)=N-\sum_i N(A_i)+\sum_{ij}N(A_i\cap A_j)-N(A_1\cap A_2\cap A_3) \tag{5}$$

上式中的和可以理解为分别对所有可能的 i 和所有的 i, j 对求和.

学生选取的语言数	N	$-[N(F)+N(L)+N(G)]$	$+[N(F\cap L)+N(L\cap G)$ $+N(F\cap G)]$	$-N(F\cap L\cap G)$
0	+1	0	0	0
1	+1	−1	0	0
2	+1	−2	+1	0
3	+1	−3	+3	−1

图 8.4

例 3 3 种语言都不选修的学生

如果一所学校有 100 名学生, 其中 40 名选修法语, 40 名选修拉丁语, 40 名选修德语, 20 名选修任意给定的一对语言, 10 名学生选修所有 3 种语言, 那么有多少学生没有选修任何语言?

此时, $N = 100, N(F) = N(L) = N(G) = 40, N(F \cap L) = N(L \cap G) = N(F \cap G) = 20$, 而且 $N(F \cap L \cap G) = 10$. 于是根据 (4), 没有选修任何语言的学生数是 $N(\overline{F} \cap \overline{L} \cap \overline{G}) = 100 - (40 + 40 + 40) + (20 + 20 + 20) - 10 = 30$. □

接下来, 我们把 (5) 运用到由前面 3 章发展的方法所不能解决的两个计数问题中.

例 4 互素整数

不大于 70 的正整数中, 与 70 互素的整数有多少?

设 $\mathscr{U}$ 是 1 到 70 间的整数的集合. "与 70 互素" 的意思是 "与 70 没有公因子". 70 的素因子是 2、5 和 7. 因此我们要计数不含因子 2、5 和 7 且小于 70 的整数个数. 设 A_1 是 $\mathscr{U}$ 中可以被 2 整除的整数集合, 或等价地说, $\mathscr{U}$ 中 2 的倍数的集合; A_2 是可以被 5 整除的整数; A_3 是可以被 7 整除的整数. 于是与 70 互素且小于 70 的正整数的个数等于 $N(\overline{A}_1 \cap \overline{A}_2 \cap \overline{A}_3)$. 我们发现

$$N = 70 \quad N(A_1) = 70/2 = 35 \quad N(A_2) = 70/5 = 14 \quad N(A_3) = 70/7 = 10$$

同时被 2 和 5 整除的整数就是被 10 整除的整数. 因此 $N(A_1 \cap A_2) = 70/10 = 7$. 根据类似的推理, $N(A_2 \cap A_3) = 70/(5 \times 7) = 2, N(A_1 \cap A_3) = 70/(2 \times 7) = 5, N(A_1 \cap A_2 \cap A_3) = 70/(2 \times 5 \times 7) = 1$. 所以, 根据等式 (5) 有:

$$N(\overline{A}_1 \cap \overline{A}_2 \cap \overline{A}_3) = 70 - (35 + 14 + 10) + (7 + 2 + 5) - 1 = 24$$ □

例 5 无缺损三进制序列

至少有一个 0、一个 1 和一个 2 的 n 位三进制 (0, 1, 2) 序列有多少?

设 $\mathscr{U}$ 是所有 n 位三进制序列的集合. 公式 (5) 计数所有性质都不成立的那些结果. 所以, 我们必须用所有性质都不成立的那些结果来构建这一问题的公式. 解决方案是分别定义 A_i 为没有 i 的 n 位三进制序列的集合, 其中 $i = 0, 1, 2$. (注意, 这里使用的是 A_0, A_1, A_2, 而不是 A_1, A_2, A_3.) 于是, 每个数字至少有一个的序列数是 $N(\overline{A}_0 \cap \overline{A}_1 \cap \overline{A}_2)$.

所有 n 位三进制序列的个数是 $N = 3^n$. 没有 0 的 n 位三进制序列个数就是 1 和 2 的 n 位三进制序列的个数. 因此 $N(A_0) = 2^n$. 类似地, $N(A_1) = N(A_2) = 2^n$. 没有 0 和 1 的三进制序列就是只有 2 的 n 位序列. 所以 $N(A_0 \cap A_1) = 1$, 同样 $N(A_1 \cap A_2) = N(A_0 \cap A_2) = 1$. 最后, 不存在没有 0, 1, 2 的三进制序列. 因此, 根据 (5),

$$N(\overline{A}_0 \cap \overline{A}_1 \cap \overline{A}_2) = 3^n - (2^n + 2^n + 2^n) + (1 + 1 + 1) - 0 = 3^n - 3 \times 2^n + 3$$ □

注意, 第 6 章使用多项式代数对计数问题建模, 第 7 章使用递推关系, 而这里利用集合论模型. 这一方法不像其他模型那样, 它不排除组合枚举. 我们仍必须解

决寻找 $N(A_i), N(A_i \cap A_j)$ 等诸如此类的子问题, 但是这些子问题都是很容易解决的问题.

有时候, 我们对 Venn 图给出非标准的输入数据. 下面的例子说明, 对于这样的情况, 如何确定 Venn 图中所有可能子集的大小.

例 6 非标准限制

假设一所学校有 100 名学生, 其中各有 40 名学生选修法语、拉丁语和德语. 20 名学生只选修法语, 20 名学生只选修拉丁语, 15 名学生只选修德语. 另外, 10 名学生选修了法语和拉丁语. 有多少名学生选修了所有 3 种语言? 又有多少学生没有选修任何语言?

我们画出这个问题的 Venn 图, 并如图 8.5 所示, 标出每个区域的编号. 设 N_i 是区域 i 中的学生数量, 其中 $i = 1, 2, \cdots, 8$. 只选修法语的学生的是子集 $F \cap \overline{L} \cap \overline{G}$, 即区域 1; 所以 N_1=20. 类似地, 根据所给的其他信息, 我们知道 $N_5 = 20, N_7 = 15$. 选修法语和拉丁语的学生是子集 $F \cap L = (F \cap L \cap \overline{G}) \cup (F \cap L \cap G)$, 即区域 2 和区域 3. 所以 $N_2 + N_3 = 10$. 选修法语的学生集合是 F, 它是由区域 1, 2, 3, 4 组成的. 所以:

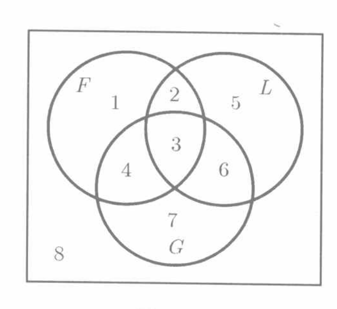


图 8.5

$$40 = N(F) = N_1 + (N_2 + N_3) + N_4 = 20 + (10) + N_4 \quad \text{或} \quad N_4 = 10$$

类似地, 集合 L 是由区域 2, 3, 5, 6 组成的, 所以 $40 = N(L) = (N_2+N_3)+N_5+N_6 = (10) + 20 + N_6$, 因此 $N_6 = 10$.

因为集合 G 是由区域 3, 4, 6, 7 组成的, 且已知 $N_7 = 15$, 以及 $N_4 = N_6 = 10$, 于是有:

$$40 = N(G) = N_3 + N_4 + N_6 + N_7 = N_3 + 10 + 10 + 15 \quad \text{或} \quad N_3 = 5$$

但是区域 3 是选修所有 3 种语言的学生子集 $F \cap L \cap G$. 因此, 有 5 名学生选修所有 3 种语言.

攻克这些问题的一般线索是把 Venn 图分解成如图 8.5 所示的 8 个区域. 对于已知大小的每个子集, 写出在这个子集中的区域 i 的 N_i 和. 通过把些方程结合起来 (有时需要联立求解它们), 最终可以确定每个区域的元素个数. 于是, 任意子集的大小就等于这个子集中各区域大小的和. 例如, 在前面的过程中, 除了 N_2 和 N_8 之外, 我们确定了所有 N_i. 但是, $N_2 + N_3 = 10$, 而且已知 N_3 等于 5, 所以 $N_2 = 5, N_8 = N(\overline{F \cap L \cap G})$ 是没有选修任何语言的学生数量. 因为所有区域总和等于 N, 所以有

$$N_8 = N - \sum_{i=1}^{7} N_i = 100 - (20+5+5+10+20+10+15) = 100 - 85 = 15 \quad \square$$

注意, 因为有 8 个基本变量 (8 个基本区域的大小), 所以解决这样问题需要 8 条信息.

练 习 8.1

练习概况 前 30 道练习都与本节的例子类似. 后 6 道练习涉及更复杂的 Venn 图的讨论, 参考例 6 的最后一段的说明来对这些问题求解.

1. (可重复地) 使用 26 个字母可以形成多少以元音开始或以元音结束的 6 字母 "单词"?

2. 有多少个 9 位数字社会保险号码 (数字可重复)?

3. 有多少至少有一对连续数字相同的 n 位三进制序列?

4. 投掷一枚硬币 9 次时, 至少出现两个正面 (不必是连续的) 的概率是多少?

5. 一手 6 张牌至少有一对 (有可能是两对, 同一种点数的 3 张牌, 满堂彩, 或同一种点数的 4 张牌) 的概率是多少?

6. 如果把不同身高的 n 个人排成一队, 那么至少有一个人正好在一个更高的人后面的概率是多少?

7. 有多少个正好有一个 8 且没有数字正好出现两次的 4 位数 (包含以 0 开头的 4 位数)?

8. 假设有 50%的家庭有洗碗机, 30%的家庭有割草机, 20%的家庭二者都有. 这两种设备都没有的家庭占家庭总数的百分比是多少?

9. 在 600 个家庭中, 150 个家庭没有孩子, 200 个家庭只有男孩, 200 个家庭只有女孩, 有多少个家庭既有男孩又有女孩?

10. 数字 $0, 1, 2, \cdots, 9$ 的排列中, 不以 8 结尾也不以 3 开头的排列有多少个?

11. 假设书架上有 200 本书, 70 本是法语书, 100 本是数学书. 有多少本不是法语也不是数学的书, 如果:

(a) 有 30 本法语数学书? **(b)** 有 60 本法语非数学书?

12. 26 个不同字母的排列中满足下列条件的有多少个:

(a) 或者包含序列 "the", 或者包含序列 "aid"?

(b) 既不包含序列 "the", 也不包含 "aid"?

13. 如果从 1 到 1 000 之间挑出一个数, 这个数或者可以被 5 整除或者是偶数 (或者二者同时成立) 的概率是多少?

14. INVITING 的字母排列中, 3 个 I 连续或者 2 个 N 连续 (或者二者同时成立) 的排列有多少个?

15. 一所学校有 200 名学生, 选修三角、概率和编织的学生各有 80 名. 选修任意一对课程的有 30 名学生, 15 名学生选修所有 3 门课程.

(a) 有多少名学生没有选修任何课程? **(b)** 有多少名学生只选修概率?

16. 假设整个学院的教授中, 有 60%喜欢网球, 65%喜欢桥牌, 50%喜欢棋类, 45%喜欢其中两种娱乐活动.

(a) 如果告知你有 20%的教授喜欢所有 3 种娱乐, 你会怀疑吗?

(b) 喜欢所有 3 种娱乐的人的最小百分比是多少?

17. 1 到 30 之间, 有多少个与 30 互素的数?

18. 1 到 280 之间, 有多少个与 280 互素的数?

19. 有多少种方法把 20 个人分配到 3 个不同的房间且每个房间至少有 1 个人?

20. 有多少个 n 位数, 其中数字 1, 2, 3 至少缺少一个? [提示: 这是一个并集的问题, 求 $N(A_1 \cup A_2 \cup A_3)$.]

21. MURMUR 的字母排列中, 任意一对连续字母都不相同的排列有多少个?

22. 如果把 3 对夫妇安置在一张圆桌周围就坐, 妻子和丈夫不坐在一起的概率是多少?

23. 有多少种方法从一组 15 名数学家、12 名统计学家和 10 名运筹学家中选出 10 名数学科学家组成一个委员会, 其中每个专业至少有一人在这个委员会中?

24. 求排列 6 个数 1, 2, 3, 4, 5, 6, 使得或者 1 的后面挨着 2, 或者 3 的后面挨着 4, 或者 5 的后面挨着 6 的方法数. [提示: 这是一个并集的问题.]

25. 有多少种方法分发一手 6 张牌, 使得至少包含一张 J、一张 8 和一张 2?

26. 有多少种方法排列单词 MISSISSIPPI 的字母, 使得或者所有 I 都是连续的, 或者所有 S 是连续的, 或者所有 P 是连续的?

27. TAMELY 的字母排列中, 或者 T 在 A 之前, 或者 A 在 M 之前, 或者 M 在 E 之前的排列有多少个? "之前" 的意思是前面任意的地方, 不一定紧挨着. [提示: 这是一个并集的问题.]

28. MATHEMATICS 的字母排列中, 两个 T 都在两个 A 之前, 或者两个 A 都在两个 M 之前, 或者两个 M 都在 E 之前的排列有多少个? 其中 "之前" 的意义与练习 27 相同.

29. Bernsteins、Hendersons 和 Smiths 这 3 对夫妇都有 5 个孩子. 如果这 3 个家庭的 15 个孩子扎营在 5 个不同的帐篷里, 每个帐篷里有 3 个孩子, 这 15 个孩子被随机分配到这 5 个帐篷, 每个家庭至少有 2 个孩子在同一个帐篷里的概率是多少?

30. 假设所有看报人中有 45%喜欢葡萄酒, 60%喜欢橙汁, 55%喜欢茶. 假设 35%喜欢其中的两种饮料, 25%喜欢所有 3 种饮料.

(a) 只喜欢葡萄酒的读者的百分比是多少?

(b) 正好喜欢 3 种饮料中的 2 种的读者的百分比是多少?

31. 假设 40 个玩具机器人中, 有 28 个机器人破损了一只轮子, 或者生锈了, 但不同时出现两种问题, 有 6 个完好的, 破损一只轮子的玩具数等于生锈的玩具数. 生锈的玩具数有多少?

32. 假设一所学校有 n 名学生, 它为学生提供 PASCAL 和 BASIC 两种程序设计语言课程. 如果有 30 名学生没有选修任何语言, 70 名学生并非只选修 PASCAL(即, 或者他们不选修 PASCAL 或者他们选修两种语言), 80 名学生并非只选修 BASIC, 20 名学生选修两种语言. 确定 n.

33. 假设一所学校有 120 名学生, 这所学校开设瑜珈和空手道课程. 如果只选修瑜珈的学生数量是选修空手道 (可能选修了空手道和瑜珈) 的人数的两倍, 如果没有选修任何技能的学生比选修两种技能的学生多 25 人, 如果 75 名学生至少选修一种技能, 那么有多少学生学习瑜珈?

34. 对 150 名大学生的调查显示: 83 人有小汽车, 97 人有自行车, 28 人有摩托车, 53 人有小汽车和自行车, 14 人有小汽车和摩托车, 7 人有自行车和摩托车, 2 人拥有所有 3 种交通工具.

(a) 有多少学生只有自行车?　　　(b) 有多少学生拥有汽车和摩托车, 但没有自行车?

35. 假设一次野餐的 150 个人当中, 90 个人带来了沙拉或三明治, 80 个人带来了三明治或奶酪, 100 个人带来沙拉或奶酪, 50 个人带来奶酪以及或者沙拉或者三明治 (或者两者都有), 60 个人至少带来两种食物, 20 个人带来所有 3 种食物. 有多少个人只带来沙拉? 有多少个人只带来三明治?

36. 在一个有 30 个孩子的班级里, 20 个孩子选修拉丁语, 14 个孩子选修希腊语, 10 个孩子选修希伯来语. 如果没有孩子选修所有 3 种语言, 而且有 8 个孩子没有选修任何语言, 有多少个孩子选修了希腊语和希伯来语? [提示: 使用公式 (5) 确定表达式 $N(L\cap G)+N(L\cap H)+N(G\cap H)$ 的值.]

8.2　容 斥 公 式

本节, 我们把计数公式 $N(\overline{A}_1\cap\overline{A}_2\cap\overline{A}_3)$ 推广到 n 个集合 $A_1,A_2,\cdots,A_n$ 上. 为了简化记法, 我们将省去表达式中的交集符号 "∩", 并把相交的集合写成积的形式. 例如, 把 $A_1\cap A_2\cap A_3$ 写成 $A_1A_2A_3$. 使用这种新记法, 不在 $A_1,A_2,\cdots,A_n$ 的任意一个集合中的元素数量可以写成 $N(\overline{A}_1\overline{A}_2\cdots\overline{A}_n)$. 下面的公式就是著名的容斥公式, 之所以这样称这个公式, 是因为它连续地包含 (加入) 和排除 (减去) 各 k 组集合的交集.

定理 1　容斥公式

设 $A_1,A_2,\cdots,A_n$ 是有 N 个元素的全集 $\mathscr{U}$ 的 n 个集合. 设 S_k 表示各 A_i 的所有 k 组集合的交集的大小之和. 于是

$$N(\overline{A}_1\overline{A}_2\cdots\overline{A}_n)=N-S_1+S_2-S_3+\cdots+(-1)^kS_k+\cdots+(-1)^nS_n \tag{1}$$

证明　为了明确各 S_k 的定义, $S_1=\sum\limits_i N(A_i), S_2=\sum\limits_{ij}N(A_iA_j), S_k$ 是 k 个 A_j 的所有交集的 $N(A_{j1}A_{j2}\cdots A_{jk})$ 的和, 最后, $S_n=N(A_1A_2\cdots A_n)$. 我们利用前面一节证明 $N(\overline{F}\ \overline{L}\ \overline{G})$ 的方法证明这一公式: 证明 (1) 的净效应是不在所有 A_i 中的元素计数一次, 而在一个或多个 A_i 中的元素计数 0 次.

如果一个元素不在所有的 A_j 中, 即, 在 $\overline{A}_1\overline{A}_2\cdots\overline{A}_n$ 中, 那么在 (1) 的右侧, 它在项 N 中计数一次, 而且在所有 S_k 中都没有计数. 所以对于 $\overline{A}_1\overline{A}_2\cdots\overline{A}_n$ 中每个元素的计数是我们希望的 1. 只在某个 A_j 中的元素在 N 中计数一次, 由 S_1 减去一次 (因为它在某个 A_j 中), 而在其他的 S_k 没有计数, 所以其计数是 $1-1=0$. 现在, 更一般地, 我们证明正好在 m 个 A_j 中的某个元素 x 在 (1) 中的净计数是 0. 元素 x 在 N 中计数一次, 由 S_1 计数 m 次 (因为 x 在 m 个 A_i 中), 由 S_2 计数 $C(m,2)$ 次 [因为, 对于 m 个包含 x 的集合中的两个集合 A_i 和 A_j, x 在交集 A_iA_j 中, 共有 $C(m,2)$ 个这样的对], ……, 一般地, 对于 $k\leqslant m$, 它由 S_k 计数 $C(m,k)$

次. 当 $h > m$ 时, 在 S_h 中对它没有计数. 所以 x 在 (1) 中的净计数是:

$$1-\binom{m}{1}+\binom{m}{2}-\binom{m}{3}+\cdots+(-1)^k\binom{m}{k}+\cdots+(-1)^m\binom{m}{m} \tag{2}$$

可以用下面的二项展开式估算上面这个二项式系数的交差和:

$$(1+x)^m=1+\binom{m}{1}x+\binom{m}{2}x^2+\cdots+\binom{m}{k}x^k+\cdots+\binom{m}{m}x^m \tag{3}$$

在 (3) 中设 $x=-1$, (3) 的右边就是 (2) 中的展开式, 而 (3) 的左边变成 $[1+(-1)]^m=0^m=0$. 因此, (2) 等于 0. ◇

推论 设 $A_1, A_2, \cdots, A_n$ 是全集 $\mathscr{U}$ 中的集合. 那么

$$N(A_1\cup A_2\cup\cdots\cup A_n)=S_1-S_2+S_3-\cdots+(-1)^{k-1}S_k+\cdots+(-1)^{n-1}S_n \tag{4}$$

证明 把公式 (1) 写成:

$$N(\overline{A}_1\overline{A}_2\cdots\overline{A}_n)=N-[S_1-S_2+S_3-\cdots+(-1)^{n-1}S_n] \tag{5}$$

接下来, 观察发现, 不在所有这些集合中的元素的数量等于总的元素数量减去在一个或多个集合中的元素的数量, 即

$$N(\overline{A}_1\overline{A}_2\cdots\overline{A}_n)=N-N(A_1\cup A_2\cup\cdots\cup A_n) \tag{6}$$

比较 (5) 和 (6), 我们看到 (5) 右边括号中的表达式是 $N(A_1\cup A_2\cup\cdots\cup A_n)$. ◇

在给出容斥公式的例子之前, 我们要强调关于运用这一公式的一个重要的逻辑要点. 为了在计数问题中使用这一公式, 我们必须选择一个全集 $\mathscr{U}$ 以及这个全集的一系列子集 A_i, 使得要计数的结果是不在所有的 A_i 中的 $\mathscr{U}$ 的元素子集. 即, 被计数的结果不满足 A_i 表示的性质.

例 1 不缺少花色的手牌

有多少种方法从标准的 52 张牌中选出一手 5 张牌, 使得对于每种花色, 这手牌至少有一张牌?

全集 $\mathscr{U}$ 应该是所有 5 张手牌的集合. 我们需要定义集合 A_i, 使得每种花色至少有一张的一手牌不在任何的 A_i 中. 稍微思考一下, 我们看到, 某种花色至少有一张等价于不缺少这种花色. 因此, 我们设 A_1 是缺少黑桃的所有 5 张手牌的集合; A_2 是缺少红心的 5 张手牌集合; A_3 是缺少方片的 5 张手牌集合; A_4 是缺少梅花的 5 张手牌集合. 现在, 这一问题是要求 $N(\overline{A}_1\overline{A}_2\overline{A}_3\overline{A}_4)$, 我们可以使用容斥公式.

接下来, 我们必须计数 N, S_1, S_2, S_3, S_4. 正如在第 5 章所提示的那样, 一手 5 张牌本身就是一个有 5 张牌的子集, 所以 $N=C(52,5)$. 缺少黑桃的手牌集合 A_1 的大小就是从 $52-13=39$ 张牌中选出没有黑桃的 5 张手牌的数量. 所以 $N(A_1)=C(39,5)$. 类似地, $N(A_i)=(39,5), i=2,3,4$, 所以 $S_1=4\times C(39,5)$. A_1A_2

中的一手牌是从 26 张非黑桃、非红心牌中选出的手牌, 所以 $N(A_1A_2) = C(26,5)$. 4 个集合中选出 2 个集合一共有 $C(4,2) = 6$ 种不同的交集, 所以, $S_2 = 6 \times C(26,5)$. 4 个集合中选出 3 个集合, 一共有 $C(4,3) = 4$ 种不同的交集, 每个集合都有 $C(13,5)$ 个手牌. 所以, $S_3 = 4 \times C(13,5)$. 最后, 一手牌中不可能所有花色为空, 所以 S_4=0. 因此, 根据 (1),

$$N(\overline{A}_1\overline{A}_2\overline{A}_3\overline{A}_4) = \binom{52}{5} - 4\binom{39}{5} + 6\binom{26}{5} - 4\binom{13}{5} + 0 \qquad \square$$

一般地, 我们说 S_k 是 n 个 A_i 的 $C(n,k)$ 个不同 k 元交集的和.

例 2　有一个空盒子的分配

有多少种方法把 r 个不同的对象分配到 5 个不同的盒子里且至少有一个空盒子?

在这个问题中, 我们不必确定 $N(\overline{A}_1\overline{A}_2\overline{A}_3\overline{A}_4\overline{A}_5)$, 因为这个问题不关心某个性质对所有盒子都不成立的那些结果. 相反, 这是一个并集的问题, 需要利用推论的公式.

设 $\mathscr{U}$ 是把 r 个不同的对象分配到 5 个盒子里的所有分配的集合. 设 A_i 是盒子 i 为空的分配集合. 于是, 至少一个盒子为空的分配数是 $N(A_1\cup A_2\cup A_3\cup A_4\cup A_5)$. 我们有 $N = 5^r, N(A_i) = 4^r$(每个对象进入其他 4 个盒子之一的分配), $N(A_iA_j) = 3^r$, 等等. 正如刚刚提示的那样, S_k 中有 $C(5,k)$ 个子集, 其中 $k = 1,2,\cdots,5$. 因此, 根据 (4):

$$N(A_1\cup A_2\cup\cdots A_5) = S_1-S_2+S_3-S_4+S_5 = \binom{5}{1}4^r-\binom{5}{2}3^r+\binom{5}{3}2^r-\binom{5}{4}1^r+0 \qquad \square$$

容易弄错并集的问题, 它在标准的容斥问题上附以诸如“至少有一个空盒子”一类的词语, 而标准的容斥问题使用的则是诸如“每个盒子里至少有一个对象”之类的词语. 前者要求性质集合之一成立, 而后者要求每个性质成立. 另外, 后面的问题必须通过补性质才能得以解决, 例如“盒子 i 是空的”, 并确定这些补性质都不成立的所有方法. 读者必须仔细推理计数问题, 并确定是构建计数所有性质都不成立的结果的解决方案, 还是构建计数一个或多个性质成立的结果的解决方案.

另一个引起混淆的源头是 S_i 的下标和 A_i 的下标. 在例 1 中, 有时候学生定义 A_1 为缺少一种花色的所有手牌, A_2 是缺少两种花色的所有手牌, 等等. 这是错误的. A 的下标是序数. A_1 表示第一个集合 (对于例 1, 是缺少黑桃的所有手牌), A_2 是第二个集合, A_3 是第三个集合, $\cdots\cdots$. 而 S 的下标是基数. S_1 是所有单个集合的大小的和, S_2 是所有两对集合的交集的大小之和, S_3 是所有 3 个集合的交集的大小之和, $\cdots\cdots$.

例 3　整数解的上界

下面的方程有多少不同的整数解:

$$x_1 + x_2 + x_3 + x_4 + x_5 + x_6 = 20 \quad 0 \leqslant x_i \leqslant 8$$

在 6.2 节, 对于这类问题可以用生成函数来解决. 现在, 我们用容斥方法解决它. 设 $\mathscr{U}$ 是所有 $x_i \geqslant 0$ 的整数解集合, 设 A_i 是 $x_i > 8$ 的整数解集合, 或者等价地, $x_i \geqslant 9$. 于是, $0 \leqslant x_i \leqslant 8$ 的解的数量是 $N(\overline{A}_1\overline{A}_2 \cdots \overline{A}_6)$.

回想一下 5.4 节的方程的整数解公式 (例 6), 我们有:

$$N = \binom{20+6-1}{20} = \binom{25}{20}$$

为了计数 $N(A_1)$, 即 $x_1 \geqslant 9$ 的结果数, 我们认为 x_i 是从 6 种类型对象中总共选出 20 个对象的选择数. 第一种类型至少为 9 的结果, 可以通过选出 9 个第一种类型的对象, 然后从 6 种类型中无限制地选出剩余的 $20-9$ 个对象来生成. 把这一推理过程运用于所有 $N(A_i)$:

$$N(A_i) = \binom{(20-9)+6-1}{(20-9)} = \binom{16}{11}$$

通过类似的推理, 现在, 先从两种类型中各选出 9 个对象, 然后再无限制地选出剩余的 $20-9-9$ 个对象, 我们有:

$$N(A_iA_j) = \binom{(20-9-9)+6-1}{(20-9-9)} = \binom{7}{2}$$

如果一个解在 3 个或者更多 A_i 中, 那么各个 x_i 的和应该超过 20, 这是不可能的. 所以, 对于 $j \geqslant 3, S_j = 0$, 而且

$$N(\overline{A}_1\overline{A}_2 \cdots \overline{A}_6) = N - S_1 + S_2 = \binom{25}{20} - \binom{6}{1}\binom{16}{11} + \binom{6}{2}\binom{7}{2} \qquad \square$$

回想一下, 使用生成函数解决前面的问题时, 我们要寻找 x^{20} 在下面的展开式中的系数:

$$(1+x+\cdots+x^8)^6 = [(1-x^9)/(1-x)]^6 = (1-x^9)^6(1-x)^{-6}$$

x^{20} 在这个积中的系数是 $a_0b_{20} + a_9b_{11} + a_{18}b_2$, 其中 a_k 是 x^k 在 $(1-x)^{-6}$ 中的系数. 而 b_k 是 x^k 在 $(1-x^9)^6$ 中的系数. x^{20} 的系数正好是前面的 $N(\overline{A}_1\overline{A}_2 \cdots \overline{A}_6)$ 的表达式. 因子 $(1-x^9)^6 = 1 - C(6,1)x^9 + C(6,2)x^{18} \cdots$ 正是在做容斥任务, 即减去一个 x_i 至少是 9 的情况, 加回两个 x_i 至少是 9 的情况. 通过使用生成函数解决这一问题, 我们无需知道这一问题中有关容斥的任何复杂性. 生成函数自动地执行所需的容斥计算.

例 4　领取帽子

设 n 个人随机进入一间黑暗的存衣处取回他们的帽子, 那么没有人拿回自己的帽子的概率是多少?

这个概率是没有人拿到自己帽子的结果的比例. 我们的全集 $\mathscr{U}$ 是 n 个人连续选择不同的帽子的所有方法. 所以 $N=n!$. 如果 A_i 是人 i 拿到自己帽子的结果集合, 那么 $N(\overline{A}_1\overline{A}_2\cdots\overline{A}_n)$ 是我们要求的没有人取回自己帽子的结果数.

于是, $N(A_i)=(n-1)!$, 因为已知人 i 取回了自己的帽子, 所以剩下的 $n-1$ 个选择帽子的所有方法数是 $(n-1)!$. 类似地, $N(A_iA_j)=(n-2)!$, 更一般地, 对于 k 个集合的交集, $N(A_{j1}A_{j2}\cdots A_{jk})=(n-k)!$. 因为 S_k 是 $C(n,k)$ 个 k 集合交集的大小之和, 根据等式 (1) 我们得到:

$$\begin{aligned}N(\overline{A}_1\overline{A}_2\cdots\overline{A}_n)&=N-S_1+S_2+\cdots+(-1)^kS_k+\cdots+(-1)^nS_n\\&=n!-\binom{n}{1}(n-1)!+\binom{n}{2}(n-2)!+\cdots\\&\quad+(-1)^k\binom{n}{k}(n-k)!+\cdots+(-1)^n\binom{n}{n}0!\\&=\sum_{k=0}^{n}(-1)^k\binom{n}{k}(n-k)!\end{aligned}$$

因为 $\binom{n}{k}=\frac{n!}{k!(n-k)!}$, 我们看到有 $\binom{n}{k}(n-k)!=\frac{n!}{k!}$. 所以:

$$N(\overline{A}_1\overline{A}_2\cdots\overline{A}_n)=\sum_{k=0}^{n}\frac{(-1)^kn!}{k!}=n!\sum_{k=0}^{n}\frac{(-1)^k}{k!}$$

因此, 没有人拿回自己帽子的概率是:

$$\frac{N(\overline{A}_1\overline{A}_2\cdots\overline{A}_n)}{N}=n!\sum_{k=0}^{n}\frac{(-1)^k}{k!}\bigg/n!=1-1+\frac{1}{2!}-\frac{1}{3!}+\cdots+\frac{(-1)^n}{n!}\qquad(7)$$

这个交错级数是 $\sum\limits_{k=0}^{\infty}(-1)^k/k!$ 的前 $n+1$ 项, 而这又是 $x=-1$ 时的 e^x 的级数. 这个级数收敛得很快. e^{-1} 和 (7) 的差总是小于 $1/n!$. 例如, $\mathrm{e}^{-1}=0.367\ 879\cdots$, 当 $n=8$ 时, (7) 的级数等于 0.367 888(即使 $n=5$, 它也是 0.366). 因此, 除了较小的 n 外, 对所有的 n, 所求的概率实际上就是 e^{-1}. 这一答案与 n 无关. □

例 4 中所处理的问题等价于求序列 $1,2,\cdots,n$ 的所有置换中, 没有数是固定不动 (即数 i 不在第 i 个位置上) 的所有置换数的概率. 序列的这种重排列称为**乱序**(derangement). 符号 D_n 表示 n 个整数的乱序个数. 根据例 4, 我们有

$$D_n=n!\sum_{k=0}^{n}\frac{(-1)^k}{k!}\approx n!\mathrm{e}^{-1}$$

为了求 D_n 的精确值, 使用下面的简单递推关系更加容易:

$$D_n=nD_{n-1}+(-1)^n\quad n\geqslant 2$$

关于这个递推关系的导出, 参阅练习 36.

例 5 图着色

有多少种方法用 n 种颜色着色图 8.6 中的 4 个顶点 (使得有公共边的顶点有不同的颜色)?

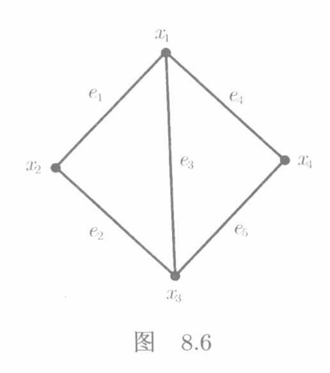


图 8.6

我们如图 8.6 那样对边 e_1, e_2, e_3, e_4, e_5 赋标签. 全集应该是着色这些顶点的所有方法的集合. 所以, $N = n^4$(每个顶点有 n 个选择). 我们必须回避的情况是用相同颜色着色相邻的顶点. 对于每一条边 e_i, 我们定义集合 A_i 是位于边 e_i 两端的顶点有相同颜色的所有着色. 于是 $N(\overline{A}_1\overline{A}_2\overline{A}_3\overline{A}_4\overline{A}_5)$ 就是 n 种颜色的正确着色数.

于是, $N(A_i) = n^3$, 因为由 e_i 连接的两个顶点得到一种颜色, 而其他两个顶点的每个顶点得到任意颜色, 所以 $S_1 = 5 \times n^3$. 类似地, $N(A_iA_j) = n^2$, 且 $S_2 = C(5,2) \times n^2$. 对于 3 个集合的交集必须稍加注意. 特别地, 边 e_1, e_2, e_3 连接 3 个顶点而不是 4 个顶点, 所以 $N(A_1A_2A_3) = n^2$(顶点 x_1, x_2, x_3 使用一种颜色, x_4 使用一种颜色). 类似地, $N(A_3A_4A_5) = n^2$. 而其他任意 3 条边连接所有 4 个顶点, 所以相应的 $N(A_iA_jA_k) = n$. 于是, $S_3 = 2 \times n^2 + [C(5,3) - 2] \times n$. 4 条边或 5 条边总是连接所有 4 个顶点, 所以 $S_4 = C(5,4) \times n$, 而 $S_5 = n$. 于是我们的问题的答案是:

$$\begin{aligned}
N(\overline{A}_1\overline{A}_2\overline{A}_3\overline{A}_4\overline{A}_5) &= n^4 - \binom{5}{1} n^3 + \binom{5}{2} n^2 \\
&\quad - \left[2n^2 + \left(\binom{5}{3} - 2\right) \times n\right] + \binom{5}{4} n - n \\
&= n^4 - 5n^3 + 8n^2 - 4n
\end{aligned}$$
□

注意上面这个展开式是这个图的色多项式 (参见 2.4 节).

本节的最后给出两个容斥公式的一般形式. 读者可以跳过这一内容.

定理 2 如果 $A_1, A_2, \cdots, A_n$ 是有 N 个元素的全集 $\mathscr{U}$ 中的 n 个集合, 那么正好包含在 m 个集合中的元素个数 N_m 和至少包含在 m 个集合中的元素个数 N_m^* 由下面式子给出:

$$\begin{aligned}
N_m &= S_m - \binom{m+1}{m} S_{m+1} + \binom{m+2}{m} S_{m+2} + \cdots + (-1)^{k-m} \binom{k}{m} S_k \\
&\quad + \cdots + (-1)^{n-m} \binom{n}{m} S_n \qquad (8)
\end{aligned}$$

$$N_m^* = S_m - \binom{m}{m-1} S_{m+1} + \binom{m+1}{m-1} S_{m+2} + \cdots + (-1)^{k-m} \binom{k-1}{m-1} S_k$$

$$+\cdots+(-1)^{n-m}\binom{n-1}{m-1}S_n \tag{9}$$

证明　N_m 的公式可以用类似于容斥公式的证明方式证明. 正好在 m 个集合中的所有元素在 S_m 中计数一次, 而在 (8) 中的其他项中没有计数. 我们必须证明包含在多于 m 个集合中的元素在 (8) 中的净计数是 0. 在这个公式中, 这个计数需要一些技巧. 对这样的技巧不感兴趣的读者可以跳过这一证明.

包含在 r 个集合中的元素在 S_k 中的计数是 $C(r,k)$, 其中 $m \leqslant r \leqslant n$, 所以这个元素在 (8) 中的净计数是:

$$\binom{r}{m}-\binom{m+1}{m}\binom{r}{m+1}+\cdots+(-1)^{k-m}\binom{k}{m}\binom{r}{k}$$
$$+\cdots+(-1)^{k-m}\binom{r}{m}\binom{r}{r} \tag{10}$$

记住, 当 $k>r$, 这个元素在 S_k 中没有计数. 由 5.5 节的例 1 可知: 从 r 个对象中选出 k 个对象, 然后再从这 k 个对象中选出 m 个特殊对象的方法数, 等于首先从 r 个对象中选出 m 个特殊对象, 然后再从余下的 $r-m$ 个对象中选出 $k-m$ 个对象的方法数. 所以有:

$$\binom{k}{m}\binom{r}{k}=\binom{r}{m}\binom{r-m}{k-m}$$

利用这个替换, (10) 就变成:

$$\binom{r}{m}-\binom{r}{m}\binom{r-m}{1}+\cdots+(-1)^{k-m}\binom{r}{m}\binom{r-m}{k-m}$$
$$+\cdots+(-1)^{k-m}\binom{r}{m}\binom{r-m}{r-m}$$
$$=\binom{r}{m}\left[1-\binom{r-m}{1}+\cdots+(-1)^{k-m}\binom{r-m}{k-m}+\cdots+(-1)^{r-m}\binom{r-m}{r-m}\right]$$

正如在定理 1 的证明中那样, 括号中的展开式正是 $x=-1$ 时 $(1+x)^{r-m}$ 的展开式. 所以 (10) 的和等于 0.

可以利用归纳法证明 N_m^* 的公式. 其要点是证明 N_m^* 的公式 (9) 满足 $N_m^*=N_m+N_{m+1}^*$: "包含在至少 m 个集合中" 的意思是 "包含在正好 m 个集合中" 或 "包含在至少 $m+1$ 个集合中". ◇

例 6　求正好有两个 1 的 4 位三进制序列的个数. 另外, 求至少有两个 1 的序列个数.

设 $\mathscr{U}$ 是所有 4 位三进制序列. 如果 A_i 是在位置 i 为 1 的 4 位三进制序列的集合, 那么 N_2 和 $N_2{}^*$ 分别是正好有两个 1 的 4 位三进制序列的个数和至少有两个 1 的 4 位三进制序列的个数. 于是, $N=3^4, S_1=C(4,1)3^3, S_2=C(4,2)3^2, S_3=$

$C(4,3)3^1, S_4 = C(4,4)3^0 = 1$. 现在, 根据公式 (8) 和公式 (9), 要求的数是:

$$N_2 = S_2 - \binom{3}{4} S_3 + \binom{3}{2} S_4 = \binom{4}{2} 3^2 - \binom{3}{2} \left[\binom{4}{3} 3^1 \right] + \binom{4}{2} 1 = 24$$

$$N_2^* = S_2 - \binom{2}{1} S_3 + \binom{3}{1} S_4 = \binom{4}{2} 3^2 - \binom{2}{1} \left[\binom{4}{3} 3^1 \right] + \binom{3}{1} 1 = 33 \qquad \square$$

细心的读者会注意到, 我们可以用简单的组合讨论直接求得 N_2.

练 习 8.2

练习概况 这些练习需要把容斥模型与第 5 章的枚举技巧 (解决子问题, 其中一些技巧相当巧妙) 结合起来. 练习 37~47 使用定理 2.

1. m 位十进制序列 (使用数字 $0, 1, 2, \cdots, 9$) 中数字 1, 2, 3 都出现的序列有多少个?

2. 有多少种方法投掷 9 枚不同的骰子, 使得所有 6 个面都出现?

3. 一手 8 张牌中至少有一张 4 的概率是多少?

4. 有多少小于等于 630 的整数与 630 互素?

5. 一手 13 张牌有下面各情况的概率是多少:

(a) 每种花色至少有一张牌? (b) 至少缺少一种花色?

(c) 每种类型的人脸牌至少有一张 (人脸牌是 A, K, Q, J)?

6. 给定 5 对不同的帽子, 5 个人每人选出两顶帽子, 有多少种方法使得没有人得到匹配的一对帽子?

7. 字母 a, a, a, b, b, b, c, c, c 的排列中, 没有 3 个连续字母相同的排列有多少个?

8. 给定 $2n$ 个字母, n 种类型中每一类型有 2 个字母, 任意连续字母都不相同的排列有多少个?

9. 26 个字母的置换中, 不包含序列 MATH, RUNS, FROM, JOE 中任何一个序列的置换有多少个?

10. 方程 $x_1 + x_2 + x_3 + x_4 = 28$ 有多少整数解满足:

(a) $0 \leqslant x_i \leqslant 10$? (b) $-10 \leqslant x_i \leqslant 20$?

(c) $0 \leqslant x_i, x_1 \leqslant 5, x_2 \leqslant 10, x_3 \leqslant 15, x_4 \leqslant 21$?

11. 有多少种方法把 24 个相同的球分配到 6 个不同的盒子, 使得前 3 个盒子中的每个盒子至多有 6 个球?

12. 所有 4 位数 (包括 0 开头的 4 位数) 中, 没有数字正好出现两次的数有多少个?

13. 一个孩子从 4 种类型的糖果中选取糖果, 如果每种糖果所取的数量都不正好是 2, 那么有多少种方法从 4 种糖果中选取 11 块糖果?

14. 圣诞老人有 n 种飞机模型, 每种有 5 个玩具飞机. 有多少种方法把在 $r(r \geqslant n)$ 个相同袜子的每个袜子里放一架飞机, 使得用到所有种类的飞机模型?

15. 一名男巫有 5 位朋友. 在一次长期的会议期间, 他在晚餐上遇到任意指定的一位朋友 10 次, 任意指定的两位朋友 5 次, 任意指定的 3 位朋友 3 次, 任意 4 位朋友 2 次, 所有 5 个朋友一次. 如果外加他自己吃 6 次晚餐, 那么确定这次会议持续了多少天.

16. 给字母表的每一个字母指定 (唯一) 一个不同的字母, 可以形成多少种密码? 利用欧拉常数 e 给出近似答案.

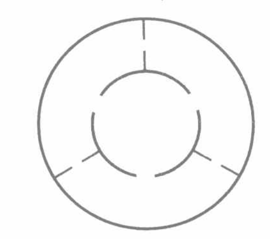
17. 有多少种方法把 10 本书分给 10 个孩子 (每个孩子一本), 然后再收回这些书, 再一次分给这些孩子, 使得他们每个人都得到一本新书?

18. $1, 2, \cdots, n$ 的排列中, 只有奇整数必须为乱序 (偶整数可以在它们的自己的位置上) 的排列有多少个?

19. 有多少种方法给数学系的 5 位教授指定秋季的两门课程 (即, 总共有 10 门不同的课程), 然后再给每位教授指定春季的两门课程, 使得没有在两个学期上同样课程的教授?

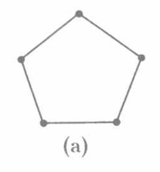
20. 考虑下面标有数字 1 到 n 的一摞 n 张卡片的游戏. 依次从 1 到 n 之间随机地选出一个不同的数, 然后从上到下拿走摞在一起的卡片, 包括写有这个数的那张卡片, 直到卡片全部拿走为止. 如果所选出数的卡片已经移去, 那么重新选出另一个数. 在某个阶段, 所选出的数是这一摞卡片中顶部卡片的近似概率是多少? (提示: 使用重复排列的方法.)

21. 有 15 名学生, 5 所不同的高中各有 3 人. 有 5 位招生官员, 分别来自 5 所大学. 每位官员依次选出 3 名学生去他们的大学. 有多少种方法这样做, 使得没有官员挑出的 3 名学生来自同一所高中?

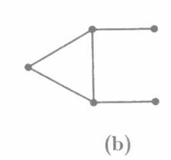


22. 用 8 种颜色给如右图所示的圆形房子中的房间涂色, 使得有公共门的房间必须有不同的颜色. 有多少种方法这样做?

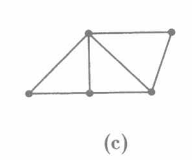
23. 有多少种方法使用 n 种颜色着色下图中的顶点, 使得相邻顶点有不同的颜色?



(a)



(b)



(c)

24. 一个房间的 4 面墙和天花板可以用 5 种颜色涂色. 如果这个房间的邻接面必须有不同的颜色, 有多少种方法这样做?

25. **(a)** 1 到 n 的整数的排列中, 对于 $i = 1, 2, \cdots, n-1$, i 后面不紧跟着 $i+1$ 的排列有多少个?

(b) 证明你的答案等于 $D_n + D_{n-1}$.

26. 考虑 n 后面紧跟着 1 的情况, 重做练习 25(a).

27. 如果连续投掷 n 次两枚相同的骰子, 有多少结果序列包含所有双数 (一对 1, 2, 等等)?

28. 有多少种方法把 n 对夫妇安排在一个圆桌的周围, 使得没有夫妇彼此换着坐?

29. 如果有 n 个家庭, 每个家庭都有 5 名成员, 有多少方法把所有 $5n$ 个人安排在一个圆桌周围, 使得每个人都坐在他或她的家庭成员的旁边? (提示: 这不是一个标准的容斥问题, 但可以使用容斥讨论的方式.)

30. a, a, a, b, b, b, c, c, c 的排列中, 相邻字母不同的排列有多少个? (提示: 需要技巧, 它不是一个标准的容斥问题.)

31. 假设一个人有 7 个朋友, 他连续一周 (7 天) 的每晚邀请不同的一组 3 个朋友参加晚宴. 有多少种方法可以这样做, 使得每个人至少被邀请一次?

32. 有 6 名网球选手, 一个月 (4 周) 内每一周, 这 6 名选手中不同对选手进行一次网球比赛. 有多少方法形成 4 场比赛的序列, 使得每名选手至少进行一次比赛?

33. 一家公司产生 8 款不同的运动衫. 每件运动衫是由 3 个不同部分做成的 (头部一部分, 中间一部分, 底部一部分). 每一部分有 6 种不同的颜色, 一件运动衫上的 3 部分必须是不同颜色的. 如果每种颜色至少出现在 8 种款式之一, 那么这 8 种款式有多少不同的组合?

34. 求有多少种方法给坐在一张圆桌周围的 6 个人分配 m 道不同菜中的一道菜, 使得相邻的人得到不同的菜?

35. 有多少种方法把 r 个不同对象分配到 n 个相同的盒子里, 且没有盒子是空的?

36. **(a)** 证明: D_n 满足递推关系 $D_n=(n-1)(D_{n-1}+D_{n-2})$.

(b) 把 (a) 中递推关系重写为 $D_n-nD_{n-1}=-[D_{n-1}-(n-1)D_{n-2}]$.

(c) 使用 (b) 列出 D_n 的值的列表, 直到 $n=10$.

(d) 使用 (a) 证明: $D(x)=\mathrm{e}^{-x}(1-x)^{-1}$ 是 D_n 的指数生成函数.

37. 有多少种方法把 i 个不同对象分配到 n 个不同的盒子里, 使得正好有 3 个盒子是空的? 至少有 3 个盒子为空时又如何呢?

38. 有多少种方法分发一手 6 张牌, 使得至多缺少一种花色?

39. 有多少种方法排列 INTELIGENT 中的字母, 使得至少有两对相同字母的连续对?

40. 如果从一个罐子里连续不断地拿出标有 $1,2,\cdots,n$ 的 n 个球, 如果被拿出的第 i 个球正好标有 i, 那么我们说发生一次*邂逅*. 如果随机地拿出 n 个球, 那么正好发生 k 次邂逅的概率是多少? 证明这个概率大约是 $\mathrm{e}^{-1}/k!$.

41. 证明: 把 r 个不同的球放入 n 个不同的盒子里且 m 个盒子里正好有 k 个球的方法数是

$$\frac{(-1)^n n!r!}{m!}\sum_{j=m}^{n}(-1)^j\frac{(n-j)^{r-jk}}{(j-m)!(n-j)!(r-jk)!(k!)^j}$$

42. **(a)** 如果 $g(x)$ 是 N_m(参见定理 2) 的 (一般) 生成函数. 证明 $g(x)=\sum\limits_{k=0}^{n}S_k(x-1)^k$. $g(x)$ 称为*选中多项式*(hit polynomial).

(b) 证明: $2[g(1)+g(-1)]$ 是包含在偶数个 A_i 中的元素个数.

(c) 使用 (b), 确定有偶数个 0 的 n 位三进制序列的数量. 用二项展开式的和来简化你的答案.

43. 证明:

$$S_m=\sum_{k=m}^{n}\binom{k}{m}N_m$$

44. 使用 (带有容斥的) 组合讨论证明下面等式:

(a) $\sum\limits_{k=0}^{m}(-1)^k\binom{m}{k}\binom{n-k}{k}=\binom{n-m}{n-r},\quad m\leqslant r\leqslant n$

(b) $\sum\limits_{k=0}^{m}(-1)^k\binom{n}{k}\binom{n-k}{m-k}=0$

(c) $\sum\limits_{k=m}^{n}(-1)^{k-m}\binom{n}{k}=\binom{n-1}{m-1}$

(d) $\sum\limits_{k=0}^{n}(-1)^k\binom{n}{k}\binom{n-k+r-1}{r}=\binom{r-1}{n-1}$

45. 使用定理 2 证明下面的等式：

$$\binom{n}{m} = \sum_{k=m}^{n} (-1)^{k-m} \binom{k}{m} \binom{n}{k} 2^{n-k}$$

46. 证明定理 2 中的公式 (9).

47. 在例 4 中, 设随机变量 X = 拿到自己帽子的人数. 求 $E(X)$.

8.3 限定位置和车多项式

本节考虑当特定对象只能出现在特定位置时, n 个对象的排列的特殊计数问题. 我们详细讨论包含 5 个对象的这样问题. 在这一过程中, 我们将指出如何把我们的分析扩展到任意的限定位置问题.

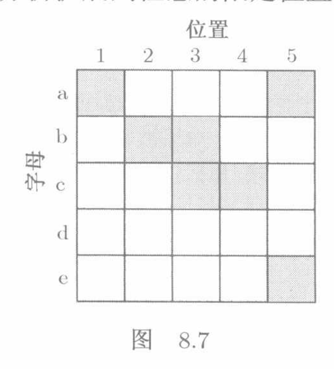


图 8.7

对于图 8.7 所示的限制条件, 考虑寻找 a, b, c, d, e 的所有排列. 即, a 不能放在位置 1 和位置 5; b 不能放在位置 2 和位置 3; c 不能放在位置 3 和位置 4; e 不能放在位置 5. 对 d 没有限制. 在图 8.7 中选出 5 个没有做标记的方格, 使得每行有一个选出方格且每列有一个选出方格, 给出一个容许排列. 例如, 容许排列 badec 对应于选出方格 (a, 2), (b, 1), (c, 5), (d, 3), (e, 4).

当把这一排列问题用 5×5 的方阵表示时, 我们可以把这一排列问题看成一个匹配问题, 即字母与位置的匹配. 在 4.4 节我们已经考虑过匹配问题. 在那里我们想要确定是否存在完全匹配. 而在此, 我们要计数有多少个完全匹配.

我们使用容斥公式, 即 8.2 节的表达式 (1), 来计数图 8.7 的容许排列个数. 设 $\mathscr{U}$ 是 5 个字母在没有限制的情况下的所有排列的集合. 所以, $N = 5!$. 设 A_i 是在位置 i 上有一个禁用字母的所有排列的集合 (注意, 我们同样还可以使用 "在禁用位置上的第 i 个字母" 来定义这些性质). 于是, 符合要求的排列数是 $N(\overline{A}_1\overline{A}_2\overline{A}_3\overline{A}_4\overline{A}_5)$. 用图 8.7 来说, A_i 是 5 个方格的所有组合的集合, 其中不同的行和列上有一个方格, 且第 i 列上的方格是黑格. $N(A_i)$ 计数的是在位置 i 放置一个禁用字母的方法数乘以在其他 4 个位置上放置剩余字母的方法数 4!(我们不担心其他字母的禁用位置). 于是, $N(A_1) = 1\times 4!, N(A_2) = 1\times 4!, N(A_3) = 2\times 4!, N(A_4) = 1\times 4!, N(A_5) = 2\times 4!$. 对各项求和, 我们得到:

$$\begin{aligned} S_1 = \sum_{i=1}^{5} N(A_i) &= 1\times 4! + 1\times 4! + 2\times 4! + 1\times 4! + 2\times 4! \\ &= (1+1+2+1+2)4! = 7\times 4! \end{aligned}$$

观察发现, (1+1+2+1+2)=7 正好是图 8.7 的黑格数. 因为每选择一个黑格 (即, 某个字母在某个禁用位置) 导致 4! 种可能性, 于是, 对任意有类似于图 8.7 的黑格的 5×5 族限定位置问题都有

$$S_1 = (\text{黑格数}) \times 4!$$

接下来, $N(A_iA_j)$ 是在位置 i 和位置 j 放置不同禁用字母的方法数乘以在其他位置放置其他字母的方法数 3!. 或者等价地, 选出两个黑格的方法数乘以 3!, 其中一个黑格在 i 列, 另一个黑格在 j 列 (且不同行). 读者应该自己检验下面的等式:

$$\begin{array}{lll} N(A_1A_2) = 1 \times 3! & N(A_1A_3) = 2 \times 3! & N(A_1A_4) = 1 \times 3! \\ N(A_1A_5) = 1 \times 3! & N(A_2A_3) = 1 \times 3! & N(A_2A_4) = 1 \times 3! \\ N(A_2A_5) = 2 \times 3! & N(A_3A_4) = 1 \times 3! & N(A_3A_5) = 4 \times 3! \\ N(A_4A_5) = 2 \times 3! & & \end{array}$$

对各项求和, 我们得到

$$S_2 = \sum_{ij} N(A_iA_j) = (1+2+1+1+1+1+2+1+4+2)3! = 16 \times 3!$$

上式中的数 16 计数从不同行和不同列选出两个黑格的方法数. 一般地, 我们得到:

$$S_k = \left(\begin{array}{c} \text{在不同行和不同列选出 } k \text{ 个} \\ \text{黑格的方法数} \end{array} \right) \times (5-k)! \tag{1}$$

因为图 8.7 中字母 d 的行没有黑格, 所以没有选出 5 个黑格的方法, 使得每个黑格在不同行 (和列) 上. 因此, S_5=0. 然而, S_3 和 S_4 仍需要我们做乏味的计算. 下面, 我们尝试探索确定从不同行和不同列选出 k 个黑格的方法数的理论.

我们可以用棋类游戏的娱乐数学问题来重述黑格选取问题. 称为 **车** (rook) 的棋子可以吃掉棋盘上位于同一行或同一列上的任意对方棋子 (前提是中间没有其他棋子). 我们不使用通常的 8×8 的棋盘, 而是在图 8.7 中由黑格组成的棋盘上玩我们的棋.

在这个黑格棋盘上放置 k 个不能互吃的车的方法数的计数, 等价于在图 8.7 中从不同行和不同列选出 k 个黑格的方法数的计数问题. 词语 "k 个不能互吃的车" 说起来更简单、更形象.

组合分析的通用技术是把大而复杂的问题拆成小且易处理的子问题. 我们将研发两个分解操作来帮助计数给定棋盘 B 上不能互吃的车.

第一个操作用于棋盘 B, 把这个棋盘分解成两个**不相交**(disjoint) 的子棋盘 B_1 和 B_2, 即, B_1 和 B_2 有不同的行和列. 通常, 我们需要重新排列棋盘才能看出子棋盘不相交的性质.

我们把图 8.7 的行和列如图 8.8 那样重新排列后, 显然在行 a 和行 e 以及列 1 和 5 上的 3 个黑格与行 b 和 c 及列 2 和 3 的 4 个黑格不相交. 设 B 是图 8.8 中黑格的棋盘, B_1 是行 a 和 e 上的 3 个黑格, B_2 是行 b 和 c 上的 4 个黑格.

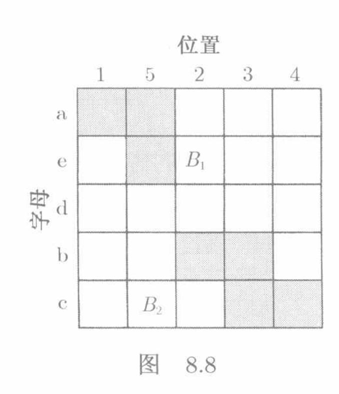


图 8.8

定义 $r_k(B)$ 为在棋盘 B 上放置 k 个不能互吃的车的方法数, $r_k(B_1)$ 是在 B_1 上放置 k 个不能互吃的车的方法数, $r_k(B_2)$ 是在棋盘 B_2 上放置 k 个不能互吃的车的方法数. 因为 B_1 有 3 个方格, 所以有 3 种方法把一个车放置在图 8.8 的棋盘 B_1 上, 类似地, 有 4 种方法把一个车放置在 B_2 上. 于是, $r_1(B_1)=3$, $r_1(B_2)=4$. 稍微想一下就可证明只有一种方法把两个车放置在棋盘 B_1 上, 有 3 种方法把两个车放置在棋盘 B_2 上, 所以 $r_2(B_1)=1$, $r_2(B_2)=3$. 注意, 因为每个子棋盘只有两行, 所以, 对于 $k \geqslant 3$, $r_k(B_1)=r_k(B_2)=0$. 对于所有棋盘定义 $r_0=1$ 很方便.

观察发现, 因为 B_1 和 B_2 不相交, 所以, 把两个不能互吃的车放置在整个棋盘 B 上有 3 种情况: 把两个不能互吃的车放置在 B_1 上 (两个车都不在 B_2 上), 或者在每个子棋盘上放置一个车, 或者把两个不能互吃的车放置在 B_2. 因此, 我们看到

$$r_2(B) = r_2(B_1) + r_1(B_1)r_1(B_2) + r_2(B_2)$$

或者, 利用事实 $r_0(B_2)=r_0(B_1)=1$,

$$r_2(B) = r_2(B_1)r_0(B_2)+r_1(B_1)r_1(B_2)+r_0(B_1)r_2(B_2) = 1\times 1+3\times 4+1\times 3 = 16 \quad (2)$$

回想一下, 上式中的数值 16 就是我们早前所说的从不同行和不同列挑出两个黑格的方法数, 也就是对所有的 $N(A_iA_j)$ 求和而得到的数.

对任意的 k 和任意可以分解成两个不相交子棋盘 B_1 和 B_2 的棋盘 B, 我们可以把导出 (2) 的推理运用于 $r_k(B)$.

引理 如果 B 是可以分解成两个不相交子棋盘 B_1 和 B_2 的黑格棋盘, 那么有

$$r_k(B) = r_k(B_1)r_0(B_2) + r_{k-1}(B_1)r_1(B_2) + \cdots + r_0(B_1)r_k(B_2) \quad (3)$$

细心的读者会注意到 (3) 非常类似于 6.2 节中两个生成函数的积的系数公式 (6). 即, 如果 $f(x) = \sum a_r x^r, g(x) = \sum b_r x^r$, 那么 x^k 在 $h(x) = f(x)g(x)$ 中的系数是 $a_k b_0 + a_{k-1}b_1 + \cdots + a_0 b_k$. 下面我们将揭示出这其间的类似性.

定义黑格棋盘 B 的**车多项式**(rook polynomial)$R(x, B)$ 为:

$$R(x, B) = r_0(B) + r_1(B)x + r_2(B)x^2 + \cdots$$

记住, 对所有 B, 有 $r_0(B)=1$. 注意, 车多项式只依赖于黑格, 而与原始分配图的大小无关. 因此, 对于上面定义的 B_1 和 B_2, 我们发现

$$R(x,B_1)=1+3x+1x^2 \quad 和 \quad R(x,B_2)=1+4x+3x^2$$

而且, 根据 (3) 与两个生成函数的积的系数公式之间的对应, 我们看到, 在整个棋盘 B 的车多项式 $R(x,B)$ 中 x^k 的系数 $r_k(B)$ 就是 x^k 在乘积 $R(x,B_1)R(x,B_2)$ 中的系数. 即

$$\begin{aligned}R(x,B)&=R(x,B_1)R(x,B_2)=(1+3x+1x^2)(1+4x+3x^2)\\&=1+[(3\times1)+(1\times4)]x+[(1\times1)+(3\times4)+(1\times3)]x^2\\&\quad+[(1\times4)+(3\times3)]x^3+(1\times3)x^4\\&=1+7x+16x^2+13x^3+3x^4\end{aligned}$$

这种乘积关系对任意的 B、B_1 和 B_2 都成立.

定理 1 如果 B 是分解成两个不相交子棋盘 B_1 和 B_2 的黑格棋盘, 那么有

$$R(x,B)=R(x,B_1)R(x,B_2)$$

并非小视生成函数在定理 1 中的作用, 然而, 我们应该知道在此使用生成函数的原因只是因为多项式乘积对应于 (3) 中给出的子棋盘合成规则. 即, 把 $r_k(B_i)$ 插入到两个多项式中, 再把它们乘起来, 这是组织 (3) 所需计算的更熟悉方法. 稍后, 我们将把棋盘 B 分解成相交的子棋盘, 而车多项式将起到真正重要的作用.

现在, 我们已经为图 8.8 中的黑格棋盘, 因此为图 8.7 中的黑格棋盘, 确定了 $R(x,B)$, 我们可以解决原来的关于 a, b, c, d, e 的容许排列问题. S_k 的表达式 (1) 可以重写成:

$$S_k=r_k(B)(5-k)!$$

根据容斥公式, 容许排列的个数是:

$$\begin{aligned}N(\overline{A}_1\overline{A}_2\overline{A}_3\overline{A}_4\overline{A}_5)&=N-S_1+S_2-S_3+S_4-S_5\\&=5!-r_1(B)4!+r_2(B)3!-r_3(B)2!+r_4(B)1!-r_5(B)0!\\&=5!-7\times4!+16\times3!-13\times2!+3\times1!-0\times0!\\&=120-168+96-26+3-0=25\end{aligned}$$

而 $r_k(B)$ 的值可以通过上面为图 8.8 所计算的车多项式 $R(x,B)$ 得到. 以容斥公式为基础的车多项式对任意限定位置的排列问题都适用.

定理 2 存在限定位置时, 排列 n 个不同对象的方法数等于

$$n!-r_1(B)(n-1)!+r_2(B)(n-2)!+\cdots+(-1)^kr_k(B)(n-k)!+\cdots+(-1)^nr_n(B)0! \quad (4)$$

其中, $r_k(B)$ 是禁用位置的棋盘 B 的车多项式 $R(x,B)$ 的系数.

我们来概括一下上面开发的小理论.

(1) 给定计数有限定位置的排列或匹配的问题, 如图 8.7, 使用矩阵表示这一限定, 其中黑格表示禁用位置.

(2) 尝试着重新排列这个矩阵, 使得黑格棋盘 B 可以分解成两个不相交的子棋盘 B_1 和 B_2.

(3) 通过观察, 确定把 k 个不能互吃的车放置在棋盘 B_i 上的方法数 $r_k(B_i)$.

(4) 使用 $r_k(B_i)$ 形成车多项式 $R(x,\ B_1)$ 和 $R(x,\ B_2)$, 并把它们乘起来得到 $R(x,B)$.

(5) 在公式 (4) 中代入 $R(x,B)$ 的系数 $r_k(B)$.

例 1 邮寄生日贺卡

有多少种方法给 3 位姑妈 A_1, A_2, A_3 和 3 位叔叔 U_1, U_2, U_3 邮寄 6 张不同的生日贺卡 $C_1, C_2, C_3, C_4, C_5, C_6$, 其中姑妈 A_1 不喜欢贺卡 C_2 和 C_4, A_2 不喜欢 C_1 和 C_5, A_3 喜欢所有贺卡, 而 U_1 不喜欢 C_1 和 C_5, U_2 不喜欢 C_4, U_3 不喜欢 C_6?

禁用位置信息如图 8.9 所示. 通过把在同一列上有黑格的行放到一起, 把同一行上有黑格的列放到一起来重组这个棋盘. 例如, 行 C_1 和 C_5 在列 A_2 和 U_1 上都有黑格, 所以我们把行 C_1 和行 C_5 放置在一起, 列 A_2 和 U_1 放置在一起; 类似地把行 C_2 和 C_4 放置在一起, 列 A_1 和 U_2 放置在一起. 我们得到如图 8.10 所示的重组后的棋盘. 因此, 原始的黑格棋盘 B 分解成两个不相交的棋盘 B_1 和 B_2, B_1 在行 C_1 和 C_5 上, B_2 在行 C_2, C_4, C_6 上. 实际上, B_2 本身又可以分成两个不相交的子棋盘 B_2' 和 B_2'', 其中 B_2'' 是就是方格 (C_6, U_3). 通过观察, 我们看到:

$$R(x,B_1)=1+4x+2x^2$$
$$R(x,B_2)=R(x,B_2')R(x,B_2'')=(1+3x+x^2)(1+x)$$

所以有

$$\begin{aligned}R(x,B)&=R(x,B_1)R(x,B_2)\\&=(1+4x+2x^2)(1+3x+x^2)(1+x)\\&=1+8x+22x^2+25x^3+12x^4+2x^5\end{aligned}$$

于是, 贺卡邮递问题的答案是:

$$\begin{aligned}&\sum_{k=0}^{6}(-1)^k r_k(B)(6-k)!\\&=6!-8\times 5!+22\times 4!-25\times 3!+12\times 2!-2\times 1!+0\times 0!\\&=720-960+528-150+24-2+0=160\end{aligned}$$ □

现在, 考虑当棋盘 B 不能分解成两个不相交子棋盘时, 确定 $R(x,B)$ 的系数的问题. 考虑图 8.11 的棋盘 B. 根据某个方格 s 是否为 k 个不能互吃的车所选择的

方格中的一个, 我们把确定 $r_k(B)$ 的问题分成两种情况. 设 B_s 是从 B 中删除方格 s 所得的棋盘, 设 B_s^* 是从 B 中删除 s 并同时删除与 s 在同一行和同一列上的所有方格所得的棋盘. 如果 s 是图 8.11 所示的方格, 那么 B_s 和 B_s^* 如图 8.12 所示. 如果不使用 s, 我们必须把 k 个不能互吃的车放置在 B_s 上. 如果使用方格 s, 那么我们必须把 $k-1$ 个不能互吃的车放置在 B_s^* 上. 因此, 我们得出结论

$$r_k(B) = r_k(B_s) + r_{k-1}(B_s^*) \tag{5}$$

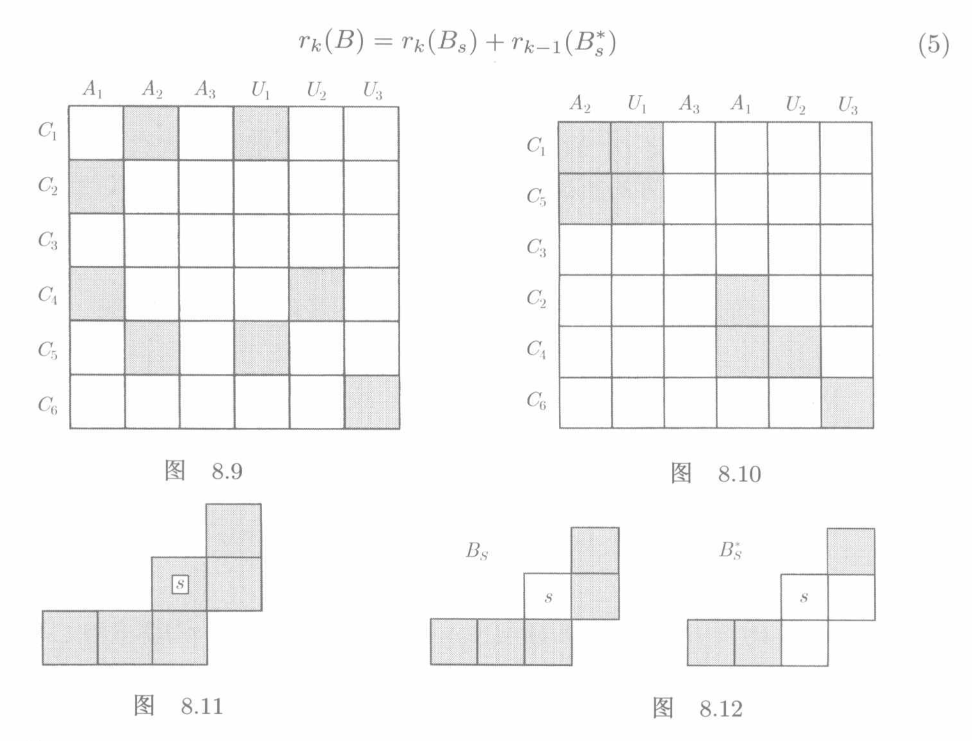


图 8.9

图 8.10

图 8.11

图 8.12

使用 7.5 节介绍的把递推关系转化成生成函数的生成函数方法, 我们从 (5) 得到

$$\begin{aligned}R(x,B) &= \sum_k r_k(B)x^k = \sum_k r_k(B_s)x^k + \sum_k r_{k-1}(B_s^*)x^k \\ &= \sum_k r_k(B_s)x^k + x\sum_h r_h(B_s^*)x^h \\ &= R(x,B_s) + xR(x,B_s^*)\end{aligned}$$

观察发现, B_s 和 B_s^* 都可以分解成不相交的子棋盘, 而通过观察容易得到这些子棋盘的车多项式:

$$\begin{aligned}R(x,B_s) &= (1+3x)(1+2x) = 1+5x+6x^2 \\ R(x,B_s^*) &= (1+2x)(1+x) = 1+3x+2x^2 \\ R(x,B) = R(x,B_s)+xR(x,B_s^*) &= (1+5x+6x^2)+x(1+3x+2x^2) \\ &= 1+6x+9x^2+2x^3\end{aligned}$$

这些结果适用于任意棋盘 B 及 B 中的任意方格 s.

定理 3 设 B 是任意黑格棋盘, s 是 B 中的一个方格, B_s 和 B_s^* 是如上定义的棋盘, 那么

$$R(x,B)=R(x,B_s)+xR(B_s^*)$$

定理 3 给出了简化任意棋盘的车多项式的方法. 如果棋盘 B_s 和 B_s^* 不能分解成两个不相交的子棋盘, 我们可以把定理 3 反复运用于 B_s 和 B_s^*. 重要的是选择尽可能分解 B 的 s.

例 2 不可分解的限制

在例 1 中, 假设叔父 U_1 的喜好发生了变化, 现在, 他不喜欢贺卡 C_2 而喜欢 C_1. 图 8.13 给出禁用位置的新棋盘. 右下角的方格 (称其为 t) 仍与其他方格不相交, 我们称它们为棋盘 B_1. B_1 不能分解成不相交子棋盘, 所以我们必须使用定理 3. 更均匀地分解 B_1 的方格 s 是如图 8.13 所示的方格 (C_2, U_1).

图 8.14 所示的棋盘 B_s 和 B_s^* 都可以分解成不相交的简单子棋盘:

$$R(x,B_s)=(1+3x+x^2)(1+3x+x^2)=1+6x+11x^2+6x^3+x^4$$
$$R(x,B_s^*)=(1+2x)(1+2x)=1+4x+4x^2$$

于是有

$$R(x,B_1)=R(x,B_s)+xR(x,B_s^*)=(1+6x+11x^2+6x^3+x^4)$$
$$+x(1+4x+4x^2)=1+7x+15x^2+10x^3+x^4$$

从而

$$R(x,B)=R(x,B_1)R(x,t)=(1+7x+15x^2+10x^3+x^4)(1+x)$$
$$=1+8x+22x^2+25x^3+11x^4+x^5$$

现在, 根据定理 2, 送发贺卡的方法数是

$$6!-8\times 5!+22\times 4!-25\times 3!+11\times 2!-1\times 1!+0\times 0!=159$$

注意, 叔父 U_1 的喜好的变化只使最终的车多项式发生一点变化: $r_4(B)$ 从 12 变成 11, $r_5(B)$ 从 2 变成 1, 而最后的答案从 160 变成 159. □

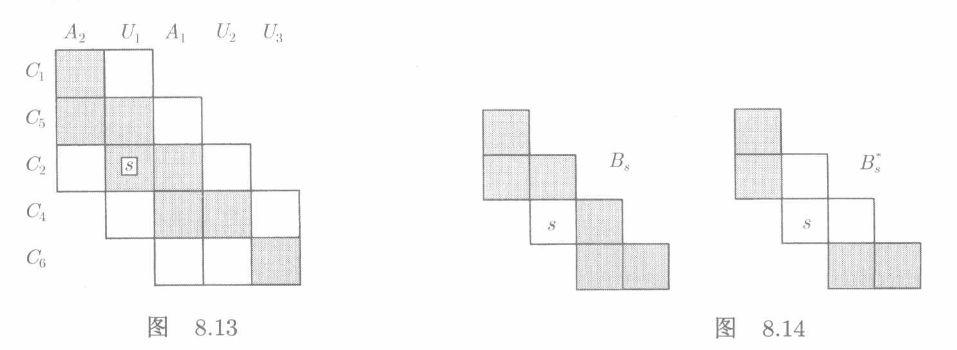


图 8.13

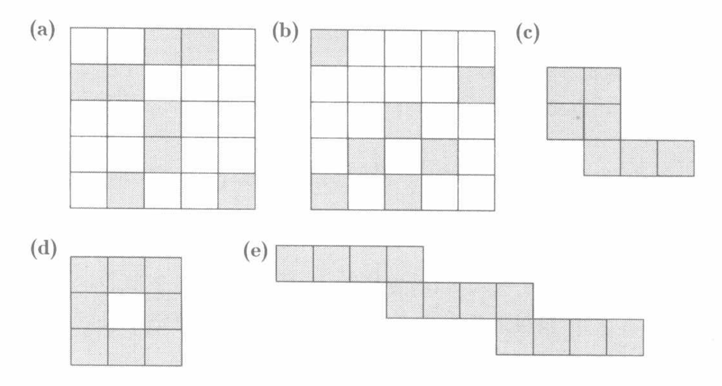
图 8.14

练 习 8.3

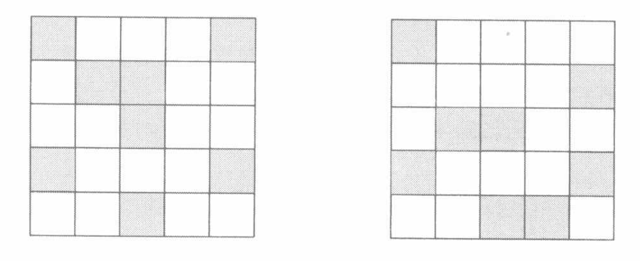
练习概况 前 9 道练习与本节的例子类似, 后面 7 道练习开发关于车多项式的理论, 以及以车多项式为基础的组合理论.

1. 为寻找 1, 2, 3, 4, 5 的所有乱排列, 描述相应的黑格棋盘.

2. 求下面棋盘的车多项式:



3. 在下面的左图给出的限制下, 求 5 名男子和 5 名女子的匹配数, 其中行代表男子, 列代表女子.



4. 在上面的右图给出的限制之下, 求 5 名男子和 5 名女子的匹配数, 其中行代表男子, 列代表女子.

5. 在一座矿山, 必须给 7 个小矮人 $D_1, D_2, D_3, D_4, D_5, D_6, D_7$ 每人分配 7 项工作 $\mathrm{J}_1, \mathrm{J}_2, \mathrm{J}_3, \mathrm{J}_4, \mathrm{J}_5, \mathrm{J}_6, \mathrm{J}_7$ 中的一项工作. D_1 不能做工作 J_1 和 J_3, D_2 不能做 J_1 和 J_5, D_4 不能做 J_3 和 J_6, D_5 不能做 J_2 和 J_7, D_7 不能做 J_4. D_3 和 D_6 可以做所有工作. 有多少种方法给这些小矮人分配不同的工作?

6. 投掷一对不同的骰子 6 次. 假设下面的有序值对 (1, 5), (2, 6), (3, 4), (5, 5), (5, 3), (6, 1), (6, 2) 都不曾出现. 第一枚骰子的 6 个值和第二枚骰子上的 6 个值在两枚骰子的 6 次投掷中都出现一次的概率是多少?

7. 计算机约会服务中心要给 4 名女子中的每个人与 5 名男子中的一个人匹配. 如果女子 1 与男子 3 和男子 5 不合适, 女子 2 与男子 1 和男子 2 不合适, 女子 3 与男子 4 不合适, 女子 4 与男子 2 和男子 4 不合适, 这 4 名女子有多种匹配?

8. 假设给 5 名官员 O_1, O_2, O_3, O_4, O_5 配置 5 辆不同的小汽车, 一辆 Escort(雅仕), 一辆 Lexus(雷克萨斯), 一辆 Nissan(尼桑), 一辆 Taurus(金牛座), 以及一辆 Volvo(沃尔沃). O_1 不驾驶 Escort 和 Nissan, O_2 不驾驶 Taurus, O_3 不驾驶 Lexus 和 Volvo, O_4 不驾驶 Lexus, O_5 不驾驶 Escort 和 Nissan. 如果随机地选取合理的汽车配置, 那么下面的概率是多少?

(a) O_1 得到 Volvo.

(b) O_2 或 O_5 得到 Volvo. (提示: 用另一个棋盘模拟这一限制.)

9. 重新排列 EERRIE 的字母, 使得没有字母出现在它的原来位置之一, 例如任何 E 都不是第 1 个、第 2 个或第 6 个字母, 计算这样形成的单词的数目.

10. 求完整 $n \times n$ 棋盘的车多项式.

11. 置换中的一个上升 (ascent) 是形如 $i, i+1$ 的连续对. 下面置换中的上升是下划线部分: $\underline{12}5\underline{34}$.

(a) 设计一个棋盘来表示 1, 2, 3, 4, 5 的置换, 使得项 (i,j) 中的对号表示在这个置换中 i 的后面紧挨着 j. 使用黑格来排除所有上升.

(b) 求 (a) 中黑格的车多项式.

(c) 求有 k 个上升的 1, 2, 3, 4, 5 的置换数. (提示: 参见 8.2 节的定理 2.)

12. 设 $R_{n,m}(x)$ 是 $n \times m$ 棋盘的车多项式 (n 行, m 列, 所有方格都可以有车).

(a) 证明: $R_{n,m}(x) = R_{n-1,m}(x) + mxR_{n-1,m-1}(x)$.

(b) 证明: $\frac{\mathrm{d}}{\mathrm{d}x}R_{n,m}(x) = nmR_{n-1,m-1}(x)$ [提示: 使用 5.5 节的等式 (5).]

13. 求有相同车多项式的两个不同棋盘 (不是一个棋盘的行或列的重新排列).

14. 考虑 $1, 2, \cdots, n$ 的所有置换中 i 既不出现在位置 i 也不出现在位置 $i+1$ 上 (n 不在位置 n 和位置 1 上) 的那些置换. 这样的置换称为一个家族 (menage). 设 $M_n(x)$ 是家族中禁用方格的车多项式. 设 $M_n^*(x)$ 是 n 可能出现在位置 1 时的车多项式, 而 $M_n^0(x)$ 是 1 和 n 都可能出现在位置 1 时的车多项式.

(a) 证明: $M_n^*(x) = xM_n^*(x) + M_n^0(x), M_n^0(x) = xM_{n-1}^0(x) + M_n^*(x), M_n(x) = M_n^*(x) + xM_{n-1}^*(x)$.

(b) 使用初始条件 $M_1^*(x) = 1 + x, M_1^0(x) = 1$, 用归纳法证明:

$$M_n^*(x) = \sum_{k=0}^{n} \binom{2n-k}{k} x^k, \qquad M_n^0(x) = \sum_{k=0}^{n-1} \binom{2n-k-1}{k} x^k$$

(c) 求 $M_n(x)$.

15. 给定 $n \times m$ 棋盘 $C_{n,m}$(参见练习 12) 和 $C_{n,m}$ 中黑格的棋盘 C, $C_{n,m}$ 中 C 的补C' 是没有涂黑的方格的棋盘.

(a) 证明:

$$r_k(C') = \sum_{j=0}^{k}(-1)^j \binom{n-j}{k-j} \binom{m-j}{k-j} (k-j)! r_k(C)$$

(b) 证明: $R(x, C') = x^n R(1/x, C)$.

16. 使用 8.2 节的定理 2, 推导计数正好有 k 个元素出现在禁用位置上的排列公式.

8.4 小结及参考文献

第 5 章所讲述的计数问题的组合复杂性通常是由诸如"每种花色至少有一张牌"这样的并存限制造成的. 本章为这样的问题构建了一个简单的集合论模型, 并彻底解决了这一模型的组合逻辑. 所得到的容斥公式被用于各种计数问题. 这一公式要求给出问题的适当集合论重述, 并给出相对直白的子问题的解决方案, 尽管如此, 这一公式消除了所有逻辑分解的负担 (以及对某些结果重复计数的担忧). 经历了第 5 章中对问题求解的无助感之后, 读者应该会很容易接受帮助他们完成大部分推理的这个强大的公式. 这就是数学的真谛.

最后一节所讨论的车多项式为在有限定位置的排列中组织容斥计算提供了优良的微型理论. 尽管在 18 世纪初, Montmort 计算了乱排列数, 而 De Moivre 也已经发表了这一公式的集合的并与交的 (非计数) 版本, 但是大约在 100 年前, J.Sylvester 才发现了容斥公式. 直到 20 世纪中期才发明了车多项式 (参见 Riordan[1]).

1. J. Riordan, *An Introduction to Combinatorial Analysis,* John Wiley & Sons, New York, 1958.

第三部分　其 他 主 题



第 9 章　Polya 枚举公式

9.1　等价和对称群

本章考查一类特殊的计数问题. 考虑用黑色、白色和红色着色正方形 4 个顶点的方法数: 这个正方形的 4 个顶点的每个顶点可以选择 3 种颜色着色, 因此给出 $3\times3\times3\times3=81$ 种不同的着色. 然而, 假设这个正方形不是在固定的位置上, 而是浮动的, 就像液体中的正方形分子一样. 现在, 又有多少种不同的顶点着色呢? 被着色的浮动图形也可能是凸 n 边形或立方体, 而且着色的可能是边或面, 而不是顶点. 浮动正方形问题等价于求 4 个被着色为黑色、白色和红色的珠子的不同 (不定向) 项链数. Polya 对计数浮动图形的不同着色公式的研究起因于化学中的一个问题, 即异构体的枚举问题.

这些问题的难点是由被着色图形的几何对称引起的. 我们开发对称集合上的特殊公式来计数图形的不同着色. 更进一步, 我们还得到一个生成函数, 它给出不同着色的**模式目录**(pattern inventory). 例如, 在所有几何对称的作用下, 立方体各角的黑白着色模式目录是:

$$b^8+b^7w+3b^6w^2+3b^5w^3+7b^4w^4+3b^3w^5+3b^2w^6+bw^7+w^8$$

其中, b^iw^j 的系数是有 i 个黑色顶点和 j 个白色顶点的不同着色的数量.

下面, 我们开发用黑色和白色着色浮动正方形 (浮动在三维空间中) 的顶点这一样本问题的公式及相关理论. 不可能画出一个浮动对象, 因为任何画面都将展示它在某个固定位置的样子. 因此, 我们从分析固定的正方形的 $2^4=16$ 种黑白着色开始. 参见图 9.1. 可以把这 16 种着色划分成着色子集, 使得当这个正方形是浮动的时候, 每个子集中的着色相互等价. 共有 6 个这样的子集 (参见图 9.1 所示的着色正方形分组), 因此有 6 种不同的浮动正方形 2 着色. 如果涉及更难的着色样本问题, 例如, 立方体的 2 着色, 或者正方形的 3 着色, 那么固定着色集合会变得非常庞大.

我们要寻找一个理论和公式来解释为什么这个正方形有 6 种不同的 2 着色. 注意, 6 个等价着色子集的大小不同.

为了定义把一个集合分成等价元素的子集的划分, 首先定义两个元素 a 和 b 等价的一般概念. 我们把这个等价写成 $a\sim b$. **等价关系**(equivalence relation) 的基本性质是:

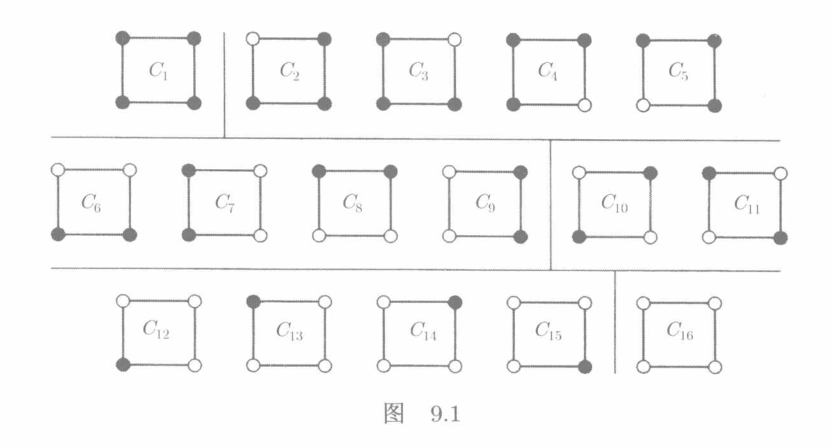


图　9.1

(i) 传递性：$a \sim b, b \sim c \Rightarrow a \sim c$.

(ii) 自反性：$a \sim a$.

(iii) 对称性：$a \sim b \Rightarrow b \sim a$.

等价的所有其他性质都可以从这 3 条性质导出. 满足这 3 条性质的任意二元关系称为等价关系. 这样的关系定义一个划分, 它把集合分成由相互等价的元素组成的子集, 这样的子集称为**等价类**(equivalence class).

例 1　等价关系

(a) 对于人的集合, 体重相同是一个等价关系, 有给定体重的所有人形成一个等价类.

(b) 对于数的集合, 差是偶数是一个等价关系, 所有偶数构成一个等价类, 所有奇数构成另一个等价类.

(c) 对于图的集合, 顶点的数量相同是一个等价关系; 对于每个 n, 有 n 个顶点的所有图形构成一个等价类. □

接下来, 我们关注把正方形映射到其自身时的运动 (参见图 9.2). 这些运动使得后面的着色等价于前面的着色. 在开发对称及其与着色等价的关系的相关理论之前, 我们仔细考察正方形及其他图形的对称.

例 2　偶凸 n 边形的对称

图 9.2 展示把正方形映射到其自身的运动集合, 即这个正方形的对称集合. 这一集合是如何得到的呢? 更一般地, 当 n 是偶数时, 凸 n 边形有什么对称呢?

这个正方形的对称可以分成两类：旋转, 即在平面内做圆周运动; 反射 (翻转), 即使用第三维的运动. 围绕正方形中心的旋转 (对称) 很容易找到. 每一个旋转是 (非零) 最小旋转的一个整数倍. 这些旋转是 π_2=90°, π_3=180°, π_4=270°, π_1=360°(或者是 0°) 的旋转.

反射 (对称) 不太容易看出来, 因为它们是在三维空间中的运动. 这些反射是：π_5= 关于垂直轴的反射, π_6= 关于水平轴的反射, π_7= 关于对顶点 a 和 c 的反

射, π_8= 关于对顶点 b 和 d 的反射.

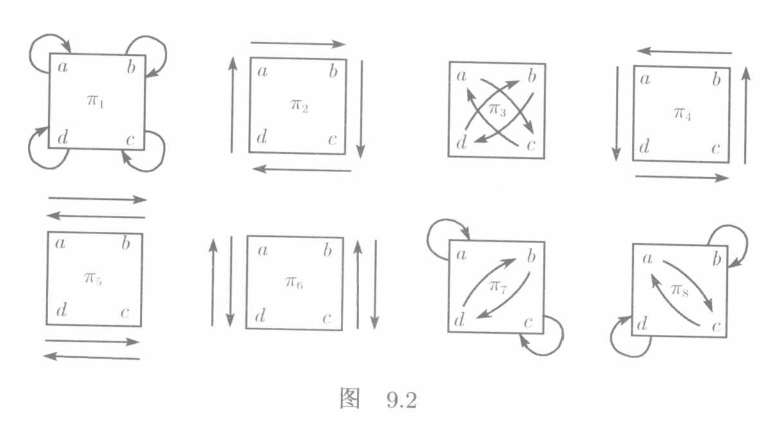


图 9.2

对于正凸 n 边形, 最小旋转是 $(360/n)°$. 这个旋转 $(360/n)°$ 的任意整数倍还是一个旋转, 所以总共有 n 个旋转. 正偶凸 n 边形有两种类型的反射: 关于两条对边的中线的反射, 以及关于两个对顶点的反射. 因为有 $n/2$ 对对顶点和 $n/2$ 对对边, 所以正偶凸 n 边形有 $n/2+n/2=n$ 个反射. 把旋转和反射加起来, 我们发现正偶凸 n 边形有 $2n$ 个对称.

我们把刚才所描述的 $2n$ 个对称各不相同的证明留作练习. 为了证明偶凸 n 边形至多存在 $2n$ 个对称, 考虑特殊顶点 x 和在顺时针方向与 x 连接的边 e. 经过一次对称运动之后, x 的位置及 e 的相对位置完全确定这个对称 (读者应该确认这一点). 一次对称可以把 x 映射到凸 n 边形的 n 个顶点中的任意一个顶点, 共有 n 种选择, 而边 e 则被映射到 x 的两侧, 共有 2 种选择. 综合起来, 凸 n 边形总共有 $2n$ 种不同的可能对称.

□

例 3　奇凸 n 边形的对称

描述五边形的对称, 更一般地, 描述奇凸 n 边形的对称.

正如在例 2 中那样, 任意正凸 n 边形有 n 个旋转对称. 五边形有 5 个旋转对称: 0°, 72°, 144°, 216°, 288°. 然而, 例 2 讨论的关于对顶点或对边的反射在五边形中是不存在的. 取而代之的是, 我们做这样的反射: 关于从一个顶点到这个顶点的对边的中点的轴的反射 (参见图 9.3). 共有 5 个这样的反射. 所以总共有 10 个对称.

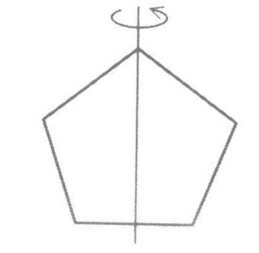


图 9.3　五边形的对称反射

在正奇凸 n 边形中, 有 n 个这样的反射, 同时, 有 n 个旋转. 把旋转和反射加起来, 我们求得正奇凸 n 边形有 $2n$ 个对称. 我们把这 $2n$ 个对称不同的证明留作练习. 另外, 利用与例 2 相同的讨论可以证明只有 $2n$ 个对称.

□

例 4　四面体的对称

描述四面体的对称.

四面体是由 6 条边和 4 个顶点相交而成的 4 个等边三角形组成的 (参见图 9.4). 除了称为 $0°$ 旋转的保持所有顶点不动的运动之外, 还可以关于一个顶点和这个顶点的对面的中心旋转 $120°$ 或 $240°$(参见图 9.4a), 或者关于对边的中点连线旋转 $180°$(参见图 9.4b). 因为有 4 组顶点及其对面和 3 对对边, 所以总共有 $1(0°$ 旋转$)+4\times2+3=12$ 个对称. 我们把这 12 个对称不同且没有其他对称的检验留作练习. □

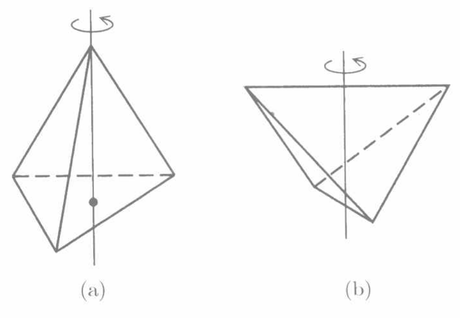


图 9.4　四面体的对称旋转

我们用正方形顶点序列的置换自然地刻画这些对称. 因此, $180°$ 旋转 π_3(参见图 9.2) 可以描述成顶点置换 $a\to c, b\to d, c\to a, d\to b$; 我们把这一置换写成 $\binom{a\ b\ c\ d}{c\ d\ a\ b}$. $90°$ 旋转 π_2 可以描述成: $a\to b, b\to c, c\to d, d\to a$, 或者 $a\to b\to c\to d\to a$.

形如 $x_1\to x_2\to x_3\to\cdots\to x_n\to x_1$ 的置换称为循环置换, 或**循环**(cycle). 因此, π_2 是长度为 4 的循环. 通常把循环写成 $(x_1x_2x_3\cdots x_nx_1)$ 的形式. 所以, $\pi_2=(abcd)$. 所有置换都可以表示成不相交循环的积 (其证明留作练习). 例如, $\pi_3=(ac)(bd)$, $\pi_4=(adcb)$, $\pi_7=(a)(bd)(c)$. 读者应该求一下其他 π_i 的循环分解. 如图 9.2 所示, 用箭头表示每一个顶点处的映射, 这样的运动描述使我们能够更加容易地描述后面计算中的置换.

在置换顶点的过程中, 对称诱发顶点着色的置换. 例如, 如果 C_i 是图 9.1 中的第 i 个正方形, 那么 π_3 就是下面的着色置换:

$$\pi_3=\begin{pmatrix} C_1 & C_2 & C_3 & C_4 & C_5 & C_6 & C_7 & C_8 & C_9 & C_{10} & C_{11} & C_{12} & C_{13} & C_{14} & C_{15} & C_{16} \\ C_1 & C_4 & C_5 & C_2 & C_3 & C_8 & C_9 & C_6 & C_7 & C_{10} & C_{11} & C_{14} & C_{15} & C_{12} & C_{13} & C_{16} \end{pmatrix} \tag{1}$$

上面置换的关键是, 尽管很容易看出对称 π_i 是如何移动这个正方形的顶点的, 但是我们真正感兴趣的是 π_i 把一种着色转变成另一种着色 (使它们等价) 的方法. 因此, 我们如下形式地定义着色等价:

着色 C 和着色 C' 等价, 记作 $C \sim C'$, 如果存在对称 π_i 使得 $\pi_i(C) = C'$ (2)

对称集合 G 的性质中, 令我们感兴趣的是使等式 (2) 中的关系 $C \sim C'$ 成为等价关系的那些性质. G 的这些性质如下 (这里的 $\pi_i \cdot \pi_j$ 意味着先执行运动 π_i, 然后再执行运动 π_j).

(1) 封闭性: 如果 $\pi_i, \pi_j \in G$, 那么 $\pi_i \cdot \pi_j \in G$; 例如, 在图 9.2 中, $\pi_2 \cdot \pi_5 = \pi_7$.

(2) 恒等元: G 包含一个恒等运动 π_{I}, 使得 $\pi_{\mathrm{I}} \cdot \pi_i = \pi_i$, $\pi_i \cdot \pi_{\mathrm{I}} = \pi_i$; 在图 9.2 中, π_{I} 是 π_1.

(3) 逆元: 对于每一个 $\pi_i \in G$, π_i 在 G 中存在逆元, 记作 π_i^{-1}, 使得 $\pi_i^{-1}\pi_i = \pi_{\mathrm{I}}$ 且 $\pi_i\pi_i^{-1} = \pi_{\mathrm{I}}$; 例如, 在图 9.2 中, $\pi_2^{-1} = \pi_4$.

观察发现, 封闭性使我们的着色关系 $\sim$ 满足传递性 [等价关系的性质 (i)]. 因为, 假设 $C \sim C'$ 且 $C' \sim C''$. 因为 $C \sim C'$, 那么一定存在某个 $\pi_i \in G$ 使得 $\pi_i(C) = C'$. 类似地, 存在某个 π_j 使得 $\pi_j(C') = C''$. 于是, 根据封闭性, 存在 $\pi_k = \pi_i\pi_j \in G$ 且 $\pi_k(C) = (\pi_i \cdot \pi_j)(C) = C''$. 因此, $C \sim C''$. 类似地, 对称的性质 (2) 和 (3) 表明我们的着色关系分别满足等价关系的性质 (ii) 和 (iii)(参见练习 16).

如果有二元运算的数学对象的集合 G 满足性质 1, 2, 3 并有结合性质 $(\pi_i \cdot \pi_j) \cdot \pi_k = \pi_i \cdot (\pi_j \cdot \pi_k)$, 则称 G 为**群**(group). 于是, 我们有下面的定理.

定理 *设 G 是集合 S(正方形的顶点) 的置换群, 而 T 是 S 的着色集合 (顶点的 2 着色). 那么, G 确定的关系 $C \sim C'$, 即存在把 C 转换成 C' 的 $\pi \in G$, 给出 T 的一个等价类划分.*

注意, S 可以是任意的对象集合, T 是可以是任意的着色集合. 下面关于群的简单引理是下节开发的计数公式的关键.

引理 *对于群 G 中的任意两个置换 π_i, π_j, 在 G 中存在唯一的置换 $\pi_k = \pi_i^{-1} \cdot \pi_j$, 使得 $\pi_i \cdot \pi_k = \pi_j$.*

证明 首先证明 $\pi_i \cdot \pi_k = \pi_j$. 因为, $\pi_k = \pi_i^{-1} \cdot \pi_j$, 所以有

$$\begin{aligned}\pi_i \cdot \pi_k &= \pi_i \cdot (\pi_i^{-1} \cdot \pi_j) = (\pi_i \cdot \pi_i^{-1}) \cdot \pi_j \quad (\text{根据结合律})\\ &= \pi_{\mathrm{I}} \cdot \pi_j = \pi_j\end{aligned}$$

接下来, 我们证明 π_k 是唯一的. 假设存在另一个置换 π_k', 使得 $\pi_i \cdot \pi_k' = \pi_j$. 那么, $\pi_i \cdot \pi_k = \pi_i \cdot \pi_k'$. 用 π_i^{-1} 乘以这个等式的两边, 我们有

$$\begin{aligned}\pi_i^{-1} \cdot (\pi_i \cdot \pi_k) = \pi_i^{-1} \cdot (\pi_i \cdot \pi_k') &\Rightarrow (\pi_i^{-1} \cdot \pi_i) \cdot \pi_k = (\pi_i^{-1} \cdot \pi_i) \cdot \pi_k'\\ &\Rightarrow \pi_{\mathrm{I}} \cdot \pi_k = \pi_{\mathrm{I}} \cdot \pi_k' \Rightarrow \pi_k = \pi_k'\end{aligned}$$ ◇

练 习 9.1

练习概况 前 13 道练习是本节给出的关于等价和对称的例子的继续. 其余练习开发与对称相关的群论的基础部分. 这些后面的练习大多需要近世代数的经验.

1. 下列关系中哪些是等价关系? 陈述你的理由.

(a) 数集合上的 $\leqslant$(小于或等于) 关系.

(b) 数集合上的 $=$(等于) 关系.

(c) 数集合上的 "差是奇数" 关系.

(d) 人的集合上的 "血缘关系".

(e) 人的集合上的 "有共同朋友" 关系.

2. 在给定运算下, 下列集合中哪些是群? 对于是群的那些集合, 哪个元素是恒等元?

(a) 非负整数 $0, 1, 2, \cdots, n$ 及其上的加法.

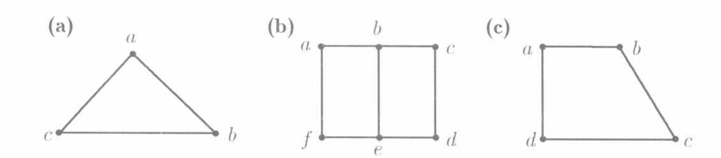
(b) 整数 $0, 1, 2, \cdots, n-1$ 及其上的模 n 加法.

(c) 所有整系数多项式及其上的多项式加法.

(d) 所有非零分数及其上的普通乘法.

(e) 所有可逆 2×2 实数值矩阵及其上的矩阵乘法.

3. 求下列各图的所有对称 (如图 9.2 所示, 用箭头表示在每一对称下所有顶点的移动):



4. 把下面的对称或置换写成循环置换的积:

(a) π_2;　**(b)** π_3;　**(c)** π_6;　**(d)** π_I;

(e) $\pi_4 \cdot \pi_7$;　**(f)** $\pi_7 \cdot \pi_4$;　**(g)** $\binom{1\ 2\ 3\ 4\ 5\ 6\ 7}{3\ 5\ 4\ 6\ 7\ 1\ 2}$.

5. 对于图 9.2 所列出的正方形的对称, 如 (1) 中那样, 给出相应的 2 着色置换:

(a) π_1;　**(b)** π_2;　**(c)** π_5;　**(d)** π_7.

6. (a) 列出三角形 3 个顶点的所有 2 着色.

(b) 对于三角形的下列对称, 如 (1) 中那样, 给出相应的 2 着色置换:

(i) π=120° 旋转;　**(ii)** π= 关于垂直轴的反射.

7. 证明: 图 9.2 所列出的正方形的 8 个对称都是不同的.

8. 证明: 例 2 和例 3 中所提到的 (偶或者奇) 凸 n 边形的所有 $2n$ 个对称都是不同的.

9. 证明: 例 4 中所列出的四面体的 12 个对称都是不同的. 证明: 四面体只有 12 个对称.

10. 求等于下面的积的正方形对称 (记住, $\pi_i \cdot \pi_j$ 表示先执行运动 π_i, 然后再执行运动 π_j):

(a) $\pi_2 \cdot \pi_4$;　**(b)** $\pi_2 \cdot \pi_5$;　**(c)** $\pi_7 \cdot (\pi_2 \cdot \pi_8)$;　**(d)** $(\pi_7 \cdot \pi_6) \cdot \pi_3$.

11. (a) 写出三角形所有对称对的积的 6×6 乘法表.

(b) 对整数 1, 2, 3, 4 作模 5 乘法, 重做 (a).

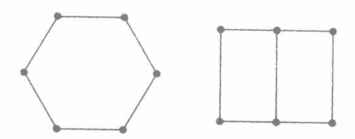
(c) 对 1, 2, 3, 4 的下面的置换群, 重做 (a).

$$\begin{pmatrix}1&2&3&4\\1&2&3&4\end{pmatrix}\quad\begin{pmatrix}1&2&3&4\\2&1&4&3\end{pmatrix}\quad\begin{pmatrix}1&2&3&4\\3&4&1&2\end{pmatrix}\quad\begin{pmatrix}1&2&3&4\\4&3&2&1\end{pmatrix}$$

12. 求正方形的两个对称 π_i, π_j, 使得 $\pi_i \cdot \pi_j \neq \pi_j \cdot \pi_i$(这表明正方形对称群不是可交换的).

13. 大多数有机混合物都是由碳原子所构成的基本结构 (碳原子是这个浮动图的顶点) 加上附着在每个碳原子上的称为基的子分子群组成的 (碳原子就像顶点, 而基就像是颜色). 假设

这样的一个有机分子有 6 个碳原子, 以及 4 个类型 A 的基和 2 个类型 B 的基. 假设这个分子有 3 种异构体, 即, 这两种类型基有 3 种不同的分布方式. 下面的两个假定的碳结构中哪一个有 3 种由 4 个 A 和 2 个 B 构成的异构体?



14. 证明: 一个等价关系可以把一个集合划分成由相互等价的元素组成的不相交子集.

15. 给出把置换分解成循环积的过程.

16. 证明: 置换群 G 的性质 (2) 和 (3) 分别蕴含定理中定义的相应等价关系的性质 (ii) 和 (iii).

17. 设 S 是一个集合, G 是 S 的置换群. 对于 S 的任意子集 S_1 和 S_2, 定义 $S_1 \sim S_2$ 表示对于某个 $\pi \in G, S_1 = \pi(S_2)(=\{\pi(s) \mid s \in S_2\})$. 证明: $\sim$ 是等价关系.

18. 证明: 正方形的 2 着色的置换的集合 [参见 (1)] 形成一个群.

19. 证明: 对于任意素数 p, 整数 1, 2, $\cdots$, $p-1$ 及其上的模 p 乘法形成一个群.

20. 如果群 G 的所有元素都可以由它的某个子集的元素的积表示, 那么称这个子集**生成群** G.

(a)下面哪些子集生成正方形的对称群?

(i) π_1, π_2, π_3; **(ii)** π_2, π_5; **(iii)** π_3, π_6; **(iv)** π_6, π_7.

(b) 证明: 正凸 n 边形的对称群可以由一个二元子集生成.

21. 如果群 G 中元素的子集 G' 本身是一个群, 那么称 G' 是 G 的一个**子群**(subgroup).

(a) 证明: $G' = \{\pi_1, \pi_2, \pi_3, \pi_4\}$ 和 $G'' = \{\pi_1, \pi_7\}$ 是正方形对称群 G 的子群.

(b) 求包含 π_3 的 G 的另一个 4 元素子群.

(c) 求正方形对称群 G 的所有子群.

22. **(a)** n 元素上有多少可能的二元关系?

(b) 有多少可能的对称二元关系?

23. 证明: $\pi_i \sim \pi_j$, 如果存在 $\pi \in G$ 使得 $\pi_i = \pi^{-1}\pi_j\pi$ 是一个等价关系.

24. 对换(transposition) 是大小为 2 的循环, 即只交换两个元素位置的置换 (而保持其他元素不动). 利用归纳法证明: n 元素的任意置换可以写成对换的合成.

25. 对于给定的 n 元素群 G, 如下定义 G 上函数 f_π: 对于每一个 $\pi' \in G, f_\pi(\pi') = \pi \cdot \pi'$.

(a) 证明: 对任意的 $\pi \in G, f_\pi$ 是一一映射.

(b) 定义 $f^*f_{\pi_2}=f_{\pi_1} \cdot f_{\pi_2}$, 证明集合$\{f_\pi\}$构成群.

9.2 Burnside 定理

现在, 我们开发计数正方形的不同 (非等价的)2 着色数目的理论. 更一般地, 在某个图形的顶点着色 (或者边或面着色) 的集合 T 中, 我们寻找由这个图的对称群 G 给出的 T 的等价类的个数 N.

假设存在一个作用于 T 中的 c 个着色上的 s 元对称群. 设 E_C 是由 C 和等价于 C 的所有着色 C' 组成的等价类. 即, 对于某个 $\pi \in G$ 使 $\pi(C) = C'$ 的所有 C'

组成的等价类. 如果 s 个 π 中的每一个都把 C 作用到不同的着色 $\pi(C)$, 那么 E_C 应该有 s 个着色. 注意, 因为 $\pi_{\mathrm{I}}(C)=C(\pi_{\mathrm{I}}$ 是恒等对称), 所以 $\pi(C)$ 的集合包含 C. 如果每一个等价类像这个等价类一样有 s 个着色, 那么

$sN=c$:(对称的数量)×(等价类的数量)=(总着色数)

求解 N, 我们有 $N=c/s$.

例如, 考虑 n 个人围绕一张圆桌的有向就坐方式 $c=n!$. 这时, 就坐方式中 $s=n$ 种是按圆周旋转的坐席, 所以, 每个等价类包含 n 种就坐方式. 因此, 等价类的数量 (非等价的围绕圆桌的就坐方式) 是 $N=n!/n=(n-1)!$

另一方面, 假设有一张有 3 个座位的小圆桌 (两个座位之间隔开 120°), 而且可以放置白椅子和黑椅子. 有 2^3=8 种把白椅子和黑椅子放置在这些位置上的方法. 参见图 9.5. 这张桌子有 3 种可能的旋转, $0°, 120°, 240°$. 我们有 $c=8$ 种 “着色” 和 $s=3$ 种对称, 但是等价类的数量不可能是 $N=8/3$, 这是一个分数!

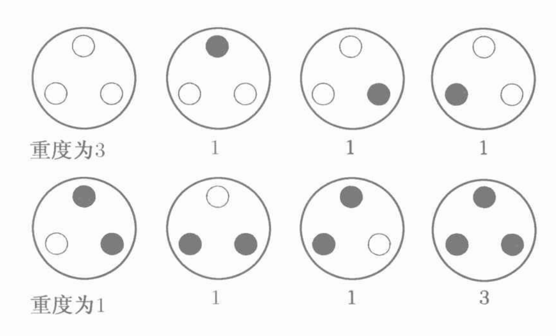


图 9.5

的确, 一把黑椅子和两把白椅子 (或反过来) 的 3 个排列形成一个等价类, 因为 $0°, 120°, 240°$ 这 3 种旋转都把一把黑椅子移到 3 个不同位置上. 然而, 任何旋转都把 3 把黑椅子 (或者 3 把白椅子) 映射到其自身, 即, 使其不变. 因此, 3 把白椅子 (及 3 把黑椅子) 的排列自身形成一个等价类. 参见图 9.5.

当一个或多个对称把一个排列映射到它自身而不是其他排列时, 我们需要修正公式 $N=c/s$ 中的分子, 在分子 c 中加入排列的重度 (重复个数). 以这种方式, 考虑排列的重度时, 每一个等价类将有 s 个成员. 因为有两个对称以及 $0°$ 对称把全黑椅子保持不变, 全白椅子也一样保持不变, 所以使用重度计数图 9.5 中的排列时, 我们的正确答案是

$$N=(3+1+1+1+1+1+1+3)/3=12/3=4$$

对正方形的 2 着色使用这种 “重度” 修正要更加复杂. 此时, 着色的等价类的大小可以是 1, 2, 4, 但不可能是 8(正方形的对称数). 一个问题是, 除恒等对称 π_{I} 之外, 其他一些 π 也可以保持某个着色 C_i 不变, 即 $\pi(C_i)=C_i$. 另一个问题是, 如

果 C_k 是 C_i 的等价类中的另一个着色, 那么可能存在若干个把 C_i 映射到 C_k 的 π. 例如, 着色 C_{10}(参见图 9.1) 在对称 $\pi_1, \pi_3, \pi_7, \pi_8$ 的作用下保持不变, 而 $\pi_2, \pi_4, \pi_5, \pi_6$ 把它映射到 C_{11}.

根据 9.1 节的引理, 把 C_{10} 映射到 C_{11} 的对称 $\pi_2, \pi_4, \pi_5, \pi_6$ 可以写成形式 $\pi = \pi_2 \cdot \pi'$ 的形式, 其中 π' 是保持 C_{11} 不变的对称. 例如,

$$\pi_5 = \pi_2 \cdot \pi_8 : \begin{pmatrix} a\ b\ c\ d \\ b\ a\ d\ c \end{pmatrix} = \begin{pmatrix} a\ b\ c\ d \\ b\ c\ d\ a \end{pmatrix} \cdot \begin{pmatrix} a\ b\ c\ d \\ c\ b\ a\ d \end{pmatrix}$$

类似地, $\pi_2 = \pi_2 \cdot \pi_1, \pi_4 = \pi_2 \cdot \pi_3, \pi_6 = \pi_2 \cdot \pi_7$. 反过来, 给定任意保持 C_{11} 不动的 π^*, $\pi_2.\pi^*$ 把 C_{10} 映射到 C_{11}, 所以, $\pi_2 \cdot \pi^*$ 一定是 $\pi_2, \pi_4, \pi_5, \pi_6$ 中的一个. 因此, 这些把 C_{10} 映射到 C_{11} 的 π 与保持 C_{11} 不变的 π 之间存在一个一一对应. 因此, 为了使用适当的重度 (例如, 因为有 4 个把 C_{10} 映射到 C_{11} 的不同 π, 所以 C_{11} 的重度为 4) 计数等价类 E 中的着色, 只要对 E 中每一个着色求出使该着色不变的 π 的数量, 然后再对这些数量求和即可.

对于由 C_{10} 和 C_{11} 构成的等价类, C_{10} 和 C_{11} 的每一个都有重度 4, 所以, 计入重度的等价类的大小是 4+4=8(= s, 即对称数), 这正是我们要求的.

一般地, 计入重度时, 每个等价类 E 有 s 个元素. 如果 $\phi(x)$ 表示保持着色 x 不变的 π 的数量, 那么, $\sum_{x\in E}\phi(x) = s$.

$\sum_{x\in E}\phi(x) = s$ 的形式证明: 设 $x_1, x_2, \cdots, x_m$ 是等价类 E 中的着色, 并设 $\pi_1, \pi_2, \cdots, \pi_s$ 是对称群. 这些 π 可以被分解成 m 组, $R_1, R_2, \cdots, R_m$, 其中, R_i 是把 x_1 映射到 x_i 的 π 的集合. 如上面 $x_1 = C_{10}, x_i = C_{11}$ 时的情况, 把 x_1 映射到 x_i 的 π 的数量等于 $\phi(x_i)$, 即是保持 x_i 不变的 π 的数量. 因此, $\sum_{x\in E}\phi(x)$ 对所有 $R_1, R_2, \cdots, R_m$ 中的 π 求和. 但是, 这些 R 的和正好等于对称数 s.

对所有等价类求和, 我们得到下面的定理, 大约在 100 年前, Burnside 首先证明了这一定理.

定理 (Burnside, 1897 年) *设 G 是集合 S(正方形的顶点) 的置换群. 设 T 是在 G 下封闭的 S 的着色集合 (顶点的 2 着色). 那么, 等价类个数 N 是*

$$N = \frac{1}{|G|}\sum_{x\in T}\phi(x)$$

或

$$N = \frac{1}{|G|}\sum_{x\in G}\psi(\pi) \qquad (*)$$

其中, $|\,G\,|$ 是置换数, $\psi(\pi)$ 是 T 中的在 π 的作用下保持不变的着色数.

"在 G 下封闭" 的意思是, 对于所有的 $\pi \in G$ 和 $x \in T, \pi(x) \in T$. 当 T 是正方形的所有顶点 2 着色集合时, 封闭性自然成立. 在 9.4 节中, 我们需要把公式 $(*)$

运用于 S 的特殊子集, 例如有两个黑色顶点和两个白色顶点这样的在 G 下封闭的特殊着色集合.

定理中的两个和都计数在某个 π 的作用下保持不变的某个着色的所有实例, 第一个是对不同的着色求和, 而第二个是对不同的 π 求和. 事实证明, 公式 (*) 在后者的计算中更有用.

下面, 我们非形式地概括公式 (*) 的要点. 正方形的所有着色的总数 c 等于 $\psi(\pi_1)$, 因为恒等对称 π_1 保持所有着色不变. 如果 8 个对称中的每一个对称把一个着色 C 映射到 8 个不同的着色, 那么在每一个等价类中我们就有 8 个着色, 因此总共有 $c/8 = \psi(\pi_1)/\mid G \mid$ 个等价类. 然而, 对于任意的着色 C, 当 π 取遍所有 G 中不同的对称时, 集合 $\pi(C)$ 的所有着色不会都不相同. 这表明每一个等价类都不可能含有 8 个元素. (*) 中的项 $\psi(\pi_i)$ 加入了重复着色的"重度", 这使得在 (*) 的和中每一个等价类都有 8 种着色.

例 1　2 色接力棒

接力棒被涂成等大小的圆柱节. 每一个圆柱可以是黑色或白色. 设想这根接力棒是在空中时, 没有上下左右之分, 这时, 如果这根接力棒有**(a)**2 个圆柱**(b)**3 个圆柱**(c)**4 个圆柱, 那么它各有多少可能的 2 着色?

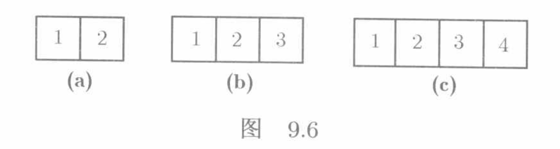


图 9.6

图 9.6 给出有 2 个圆柱、3 个圆柱和 4 个圆柱的接力棒. 与圆柱数量无关, 一根接力棒有两个对称: π_1 是这根接力棒的 $0°$ 旋转, 即是恒等对称, 而 π_2 是 $180°$ 旋转.

(a) 对于有 2 个圆柱节的接力棒, 在 π_1 的作用下保持不变的 2 着色集合是这根接力棒的所有 2 着色, 一共有 2^2=4 种 2 着色, 所以 $\psi(\pi_1)=4$. 在 π_2 的作用下保持不变的 2 着色集合是由全黑和全白着色组成的, 所以 $\psi(\pi_2)=2$. 根据 Burnside 定理, 不同的着色数等于 $\frac{1}{2}[\psi(\pi_1)+\psi(\pi_2)]=\frac{1}{2}(4+2)=3$.

(b) 对于有 3 个圆柱节的接力棒, 所有 2^3 个 2 着色在 π_1 的作用下保持不变, 所以 $\psi(\pi_1)=8$. 在 π_2 的作用下保持不变的 2 着色集合是这样的着色, 即两端圆柱节之间的圆柱节的颜色任意, 而两端的圆柱节颜色相同, 所以 $\psi(\pi_2)=2\times2=4$. 因此, 不同的着色数是 $\frac{1}{2}[\psi(\pi_1)+\psi(\pi_2)]=\frac{1}{2}(8+4)=6$.

(c) 对于有 4 个圆柱节的接力棒, 所有 2^4=16 种 2 着色在 π_1 的作用下保持不变, 所以 $\psi(\pi_1)=2^4=16$. 在 π_2 的作用下保持不变的 2 着色集合是这样的着色, 即两端圆柱节有相同的颜色, 而内部的两个圆柱节有相同的颜色, 所以 $\psi(\pi_2)=2\times2=4$. 因此, 不同的着色数等于 $\frac{1}{2}[\psi(\pi_1)+\psi(\pi_2)]=\frac{1}{2}(16+4)=10$. □

例 2　3 色接力棒

(同例 1 一样) 如果接力棒是浮动的, 那么 n 个圆柱节的接力棒有多少种不同的 3 着色?

如同例 1 一样, 接力棒的对称是 0° 和 180° 旋转. 我们把公式 (∗) 运用于作用于 n 个圆柱节 3 颜色接力棒上的 0° 和 180° 旋转的群上. 不动的接力棒有 3^n 种着色, 所以 $\psi(0°)=3^n$. 在 180° 旋转之下保持不变的着色数依赖于 n 的奇偶性.

如果 n 是偶数, 这根接力棒的一半上的 $n/2$ 个圆柱节可以是任意颜色, 有 $3^{n/2}$ 种选择, 于是对于在 180° 旋转之下保持不变的着色来说, 这个接力棒另外一半上的每一个圆柱都必须是对应颜色. 所以 $\psi(180°)=3^{n/2}$, 于是根据公式 (∗), 我们有 $N=\frac{1}{2}(3^n+3^{n/2})$.

为了枚举当 n 是奇数时在 180° 旋转之下保持不变的接力棒着色数, 我们可以对接力棒中央的这个 "奇" 圆柱节使用任意颜色, 有 3 种选择. 而对中央两侧之一的 $(n-1)/2$ 个圆柱节可以使用任意颜色, 有 $3^{(n-1)/2}$ 种选择, 而且另外一半的 $(n-1)/2$ 个圆柱节一定要着色成对称的颜色, 所以 $\psi(180°)=3\times 3^{(n-1)/2}=3^{(n+1)/2}$ 且 $N=\frac{1}{2}(3^n+3^{(n+1)/2})$. □

例 3　3 色项链

假设一条项链有黑、白和红 3 种颜色的珠子. 有多少不同的 n 珠项链?

当把 n 个珠子平放成一个圆时, 珠子占据正凸 n 边形的顶点的位置. 因此, 我们的问题是使用 3 种颜色时凸 n 边形的顶点的着色个数. 答案取决于上面所说的 "不同" 的意义是什么. 如果不允许珠子沿着项链移动, 即这个凸 n 边形是固定的, 那么此时的答案就是 $3\times 3\times\cdots\times 3=3^n$($n$ 个顶点中的每一个顶点都有 3 种颜色选择). 而我们的问题更切实的解释是允许珠子自由地绕圆移动, 即这个凸 n 边形可以自由旋转 (但此时不允许翻转). 我们用公式 (∗) 计数由凸 n 边形的旋转对称所产生的 3 着色等价类的个数 N. 现在只对 $n=3$ 给出计数. 我们将在 9.3 节开发对于更大的 n 的更一般的计数技术.

一共有 3^3=27 种 3 珠子 3 着色, 而且有 0°, 120°, 240° 三个旋转. 0° 旋转保持所有着色不变, 所以 $\psi(0°)=27$. 120° 旋转无法使某种颜色只在一个顶点处出现的着色不变. 因此 120° 旋转只保持单一着色不变, 因此 $\psi(120°)=3$. 240° 旋转是反方向的 120° 旋转, 所以 $\psi(240°)=3$. 根据公式 (∗), 我们有

$$N=\frac{1}{3}(27+3+3)=11$$

□

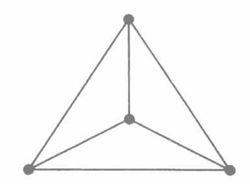
练　习　9.2

练习概况　这些练习继续使用 Burnside 定理计数例 1、例 2 和例 3 中介绍的着色.

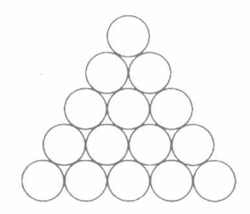
1. 使用红、白、蓝和绿 4 种珠子可以形成多少不同的 n 珠项链 (假设项链可以旋转但不能翻转)?

(a) $n = 3$;　　　　　(b) $n = 4$.

2. 有多少种方法在下面别针的 4 个顶点处放置钻石、蓝宝石或红宝石?



3. 把 15 个球如下所示放置成一个三角形. 如果排列可以自由旋转, 那么使用 3 种颜色的球可以形成多少不同的排列?



4. 有多少种方法给可以自由旋转的 8×8 的棋盘的 64 个方格进行 2 着色?

5. 使用 3 种不同类型的木马来构建旋转木马. 有多少种 5 匹马的旋转木马?

6. 多米诺骨牌是一面由两张相邻的正方形组成的木制长方形薄片 (另一面是黑色的). 每个正方形或者是空白的, 或者有 1, 2, 3, 4, 5, 6 个点.

(a) 有多少种不同的多米诺骨牌?

(b) 通过把一张多米诺骨牌模拟成从 0, 1, 2, 3, 4, 5, 6 中重复地选出两个数的无序子集来检查你的答案.

7. 由数字 0, 1, 6, 8, 9 组成的两个 n 位十进制序列是竖直相等的, 如果从竖直方向上颠倒地读取其中一个序列就会产生另一个序列, 如 0068 和 8900. 有多少不同的竖直相等的 n 位 (0, 1, 6, 8, 9) 序列?

8. 有多少种方法使用红、白、蓝和黄色给一个 (底是正方形的) 浮动金字塔的 5 个面着色?

9. **(a)** 求 3 个对象的所有可能置换的群.

(b) 求把 12 个相同球分配到 3 个相同盒子里的方法数. [**提示:** 首先设盒子不同, 然后使用 (a) 的结果.]

10. 如果接力棒的相邻圆柱必须有不同的颜色, 那么一根接力棒的 n 个圆柱节的 3 着色数是多少?

11. 如果正方形的相邻顶点必须有不同的颜色, 那么在旋转和反射之下, 这个正方形的顶点的 3 着色有多少种?

12. **(a)** 有多少种方法把 12 个红软糖分配给一对 (不可区分的) 双胞胎女孩和一对双胞胎男孩这 4 个孩子?

(b) 如果每个孩子必须至少有一个软糖, 重做问题 (a).

13. **(a)** 证明: 对于任意给定的 i, 保持着色 C_i(图 9.1 的着色) 不变的正方形的对称的子集 G_i 是群 (所有对称的一个子群).

(b) 求 G_4.　　　　　**(c)** 求 G_7.

9.3 循环指标

没有更深层次的理论支持, 我们会发现很难运用 Burnside 定理来计数浮动图的不同着色. 回想一下, 9.2 节的 Burnside 定理说的是在对称群 G 的作用下, 图着色集合 T 的等价类数量是:

$$N = \frac{1}{|G|}\sum_{\pi\in G}\psi(\pi)$$

其中, $\psi(\pi)$ 是在 π 下保持不变的 T 的着色数. 如果集合 T 是凸 10 边形或立方体的顶点的 3 着色的集合, 那么几乎无法确定 $\psi(\pi)$, 即在这个图的各个对称之下保持不变的着色数. 但是, 我们将看到根据 π 的结构, 很容易确定 $\psi(\pi)$. 我们以正方形的 2 着色为例, 开发简化 $\psi(\pi)$ 的计算的理论.

我们把 Burnside 定理运用于正方形的 2 着色. 首先, 通过观察 (使用图 9.1 和图 9.2) 确定在运动 π_i 的作用下不变的 2 着色数. 正如对每一个 π_i 计数 $\psi(\pi_i)$ 那样, 我们寻找一种模式, 它能使我们精确地预测哪一种着色在每一个 π_i 的作用下一定是不变的. 做一个表格记录 π_i 及在其作用下不变的着色会很有帮助; 参见图 9.7 的列 (i) 和 (ii)[列 (iii) 是后面要讨论的].

(i) 运动 π_i	(ii) 在 π_i 的作用下不变的着色	(iii) 循环结构表示
π_i	16— 所有着色	x_1^4
π_2	$2—C_1, C_{16}$	x_4
π_3	$2—C_1, C_{10}, C_{11}, C_{16}$	x_2^2
π_4	$2—C_1, C_{16}$	x_4
π_5	$2—C_1, C_6, C_8, C_{16}$	x_2^2
π_6	$2—C_1, C_7, C_9, C_{16}$	x_2^2
π_7	$8—C_1, C_2, C_4, C_{10}$ $C_{11}, C_{12}, C_{14}, C_{16}$	$x_1^2x_2$
π_8	$8—C_1, C_3, C_5, C_{10}$ $C_{11}, C_{13}, C_{15}, C_{16}$	$x_1^2x_2$

图 9.7

0° 旋转 π_1 保持每一个顶点不动, 因此它使所有着色不变. 所以 $\psi(\pi_1)$=16. 90° 旋转 π_2 循环地置换顶点 a, b, c, d. 在 90° 旋转下保持不变意味着: 着色的每一个顶点经 90° 旋转之后的颜色与旋转之前的颜色相同. 因为 π_2 把 a 映射到 b, 所以在 π_2 下保持不变的着色必定是 a 和 b 有相同颜色. 类似地, 这样的着色必定在 b 和 c 有相同颜色, 在 c 和 d 有相同颜色. 把这些条件放到一起, 这表明只有全黑或全白的着色 C_1 和 C_{16} 才可能在 π_2 的作用之下保持不变. 因此 $\psi(\pi_2)$=2. 一般地, 一个着色 C 在 π 的作用下不变, 当且仅当对于每一个顶点 v, C 在 v 的着色与在 $\pi(v)$ 的着色相同, 从而该对称保持 $\pi(v)$ 的颜色不变.

接下来考虑 180° 的旋转 π_3. 看一下图 9.2 中 π_3 的描述, 我们看到 π_3 使顶点 a 和 c 彼此交换, 顶点 b 和 d 彼此交换. 因此, 在 π_3 下保持不变的着色必定在 a 与 c 处有相同颜色, 在 b 与 d 处有相同颜色 (不需要其他的条件). 因为对于 a 和 c 有两种选择, b 和 d 有两种选择, 所以我们可以构建 2×2=4 种在 π_3 的作用下不变的着色, 即, $C_1, C_{10}, C_{11}, C_{16}$. 因此, $\psi(\pi_3)=4$.

对称 π_4 是 270° 或者 −90° 旋转, 所以这个对称类似于 π_2, 因此, $\psi(\pi_4) = \psi(\pi_2)=2$. 水平旋转 π_5 把顶点 a 与 b 彼此交换, c 与 d 彼此交换. 与 180° 旋转 π_3 类似, 如果顶点 a 和 b 有相同颜色, 这一共有 2 种选择, 白或黑, 同时顶点 c 和 d 有相同的颜色, 这也有 2 种选择, 那么对称 π_5 使这个着色不变. 因此, 像 π_3 一样, 我们有 $\psi(\pi_5)=2\times2=4$ 种不变的着色, 即, C_1, C_6, C_8, C_{16}.

一种模式已经浮出水面. 读者应该能够很快预测 π_6 也同样使 2×2=4 种正方形 2 着色不变, 因为这一对称使两对顶点彼此交换位置 (对换). 形式上, 对换是作用于两个元素上的循环置换. 我们枚举出的不变着色都基于这样的事实: **如果 π_i 循环地置换顶点的一个子集 (即顶点形成 π_i 的一个循环), 那么在 π_i 作用之下保持不变的任意着色使这些顶点有相同的颜色.**

正如 9.1 节所提到的那样, 任意的 π_i 可以表示成为不相交的循环之积. 例如, $\pi_3=(ac)(bd)$, $\pi_4=(adcd)$. 对于每一种对称, 我们都需要得到一个这样的循环表示, 然后再计数给顶点的每一个循环分配颜色的方法数. 为了进一步使用, 我们根据循环的长度 (循环的数目) 对它们分类. 事实证明, 对于对称 π_i, 用 x_1, x_2 等乘积形式编码 π_i 的循环信息是很方便的, 其中, 在 π_i 下, 1 个顶点的循环用一个 x_1 表示, 大小为 2 的循环用一个 x_2 表示, 等等. 这样的表示称为对称的**循环结构表示**(cycle structure representation).

π_2 和 π_4 的循环结构表示是 x_4, 因为它们都是长度为 4 的循环: $\pi_2=(abcd)$, $\pi_3=(adcb)$. π_3, π_5, π_6 的循环结构表示是 x_2x_2, 或 x_2^2, 因为它们都是由两个长度为 2 的对换组成的. 图 9.7 中的列 (iii) 给出了每一个对称的循环结构表示.

π_1 的情况如何呢? 之前, 我们只要说 π_1 保持所有着色不变就足够了. 现在, 我们要指出保持一个顶点不变的循环是长度为 1 的循环. 因此, π_1 是由 4 个长度为 1 的循环组成的. 它的循环结构表示是 x_1^4. 我们 "预测" π_1 保持 $2^4=16$ 种着色不变, 即, π_1 保持所有 2 着色不变.

对于任意的 π_j, 在其作用下保持不变的着色数可以通过设 π_j 的循环结构表示中的每个 x_j 等于 2(或更一般地, 可用颜色数) 而得到, 即,

$$\Psi(\pi) = 2^{\pi\text{中循环的数目}}$$

我们把这一理论用于 π_7 和 π_8. 因为对于这样的反射, 其循环结构表示都是 $x_1^2x_2$(根据图 9.2 的箭头), 因此, 对于每一个对称, 有 $2^2\times2=8$ 种着色不变. 注意, 因为 180° 是沿着对顶点和对边中点的翻转, 因此, 所有顶点都在大小为 2 的循环内, 除非是

在这个翻转下保持不变的顶点.

为了利用 Burnside 定理得到浮动正方形的不同 2 着色的数量, 我们把图 9.7 中的列 (ii) 中的数加起来, 并除以 8:

$$\frac{1}{|G|}\sum_{\pi\in G}\Psi(\pi)=\frac{1}{8}(16+2+4+2+4+4+8+8)=\frac{1}{8}(48)=6$$

还有得到这个答案的更简单方法. 首先, 以代数方式把每个对称的循环结构表示加起来, 合并同类项, 再除以 8. 根据图 9.7 的列 (iii), 我们得到 $\frac{1}{8}(x_1^4+2x_4+3x_2^2+2x_1^2x_2)$. 这一表达式称为对称群 G 的**循环指标**(cycle index)$P_G(x_1,x_2,\ldots,x_k)$. 令这个循环指标中的每个 x_i 等于 2, 即, $P_G(2,2,\cdots,2)$, 我们可以得到相同的结果. (之前, 这一步是反过来进行的: 我们首先令每个循环结构表示中的 x_i 等于 2, 然后再加起来.)

假设我们用 3 着色取代 2 着色. 那么可以使用相同的推理, 但是此时, 每个循环中的顶点的颜色有 3 种选择. 如果一个对称有 k 个循环, 那么它将使正方形的 3^k 个 3 着色不变, 不同的 3 着色数是 $P_G(3,3,\cdots,3)$. 更一般地, 对于任意的 $m, P_G(m,m,\cdots,m)$ 是浮动正方形的非等价 m 着色数. 用正方形的循环指标所得到的这一着色计数公式的讨论对其他有相关对称的任意集合的着色计数都适用.

定理 设 S 是元素的非空集合, 而 G 是 S 的对称群, G 的作用给出 S 的 m 着色集合上的一个等价关系. 那么 S 的非等价 m 着色数由 $P_G(m,m,\cdots,m)$ 给出.

例 1 着色项链

使用上面的定理重新解决 9.2 节例 3 的问题, 计数用黑、白、红 3 种珠子所形成的 n 珠项链数.

回想一下, 项链上的珠子可以自由地绕圆旋转, 但不允许翻转, 而且 n 个珠子的不同 3 着色链的数目等于浮动凸 n 边形的 3 着色数目. 对于 $n=3$, 旋转 0°, 120°, 240° 的循环结构表示分别是 x_1^3, x_3, x_3. 因此, $P_G=\frac{1}{3}(x_1^3+2x_3)$. 3 珠子的 3 着色链的数目等于 $P_G(3,3,3)=\frac{1}{3}(3^3+2\times3)=11$. 更一般地, 3 珠子 m 着色项链数是 $P_G(m,m,m)=\frac{1}{3}(m^3+2m)$.

下面, 我们尝试解决更复杂的情况: $n=8$(参见图 9.8). 此时旋转是 $0°,45°,90°,135°,180°,225°,270°,315°$.0° 旋转是由 8 个 1 循环组成的. 45° 旋转是由循环置换 $(abcdefgh)$ 组成的. 90° 旋转有循环分解 $(aceg)(bdfh)$. 135° 旋转是循环置换 $(adgbehcf)$. 180° 旋转有循环分解 $(ae)(bf)(cg)(dh)$. 因此, 循环结构表示是: 0° 旋转是 x_1^8; 45° 旋转是 x_8; 90° 旋转是 x_4^2; 135° 旋转是 x_8; 180° 旋转是 x_2^4. 而 225°, 270°, 315° 旋转分别是 135°, 90°, 45° 的逆旋转, 因此它们有相应的

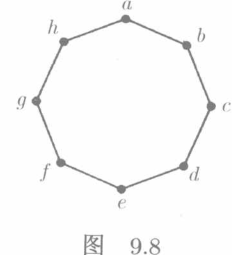


图 9.8

循环结构表示. 合并这些项, 我们得到

$$P_G = \frac{1}{8}(x_1^8 + 4x_8 + 2x_4^2 + x_2^4)$$

m 着色的不同 8 珠项链数是:

$$\frac{1}{8}(m^8 + 4m + 2m^2 + m^4)$$

对于 $m = 3$, 我们有

$$\frac{1}{8}(3^8 + 4 \times 3 + 2 \times 3^2 + 3^4) = \frac{1}{8}(6\ 561 + 12 + 18 + 81) = 834 \qquad \square$$

例 2 四面体的顶点着色

使用这一定理确定浮动四面体的 4 个顶点的 3 着色数. 在 9.1 节的例 4 中, 我们列出了四面体的 12 个对称 (参见 9.1 节的图 9.4): 0° 旋转; 120° 旋转和 240° 旋转 (一共有 8 个), 它们是关于一个顶点和这个顶点对面中心的旋转; 再加上 3 个 180° 的关于对边中线的旋转.

0° 旋转有循环结构表示 x_1^4. 120° 旋转是关于顶点 a 以及它的对面 bcd 的中心连线的旋转, 它有循环分解 $(a)(bcd)$, 所以它的循环结构表示是 x_1x_3. 根据对称性, 其他 120° 旋转和 240° 旋转都有相同的循环结构表示. 180° 旋转是关于边 ab 和 cd 的中线的旋转, 它有循环分解 $(ab)(cd)$, 所以其循环结构表示是 x_2^2. 根据对称性, 其他的 180° 旋转有相同的循环结构表示. 因此我们有

$$P_G = \frac{1}{12}(x_1^4 + 8x_1x_3 + 3x_2^2)$$

所以, 不同的顶点 3 着色数是:

$$P_G(3,3,3,3) = \frac{1}{12}(3^4 + 8 \times 3 \times 3 + 3 \times 3^2) = \frac{1}{12}(81 + 72 + 27) = 15 \qquad \square$$

练 习 9.3

练习概况 前 11 道练习计数浮动图形的不同着色. 其余问题解决相关理论. 注意, “浮动” 意指允许所有可能的旋转和反射.

1. 有多少种方法使用 4 种不同的颜色给浮动正方形的顶点着色?

2. 有多少种方法给五边形的顶点做 4 着色, 其中:

(a) 这些着色只因旋转而不同?

(b) 这些着色因旋转和反射而不同?

3. 有多少种方法给六边形的顶点做 3 着色, 其中:

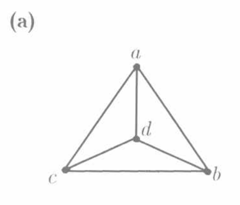
(a) 这些着色只因旋转而不同?

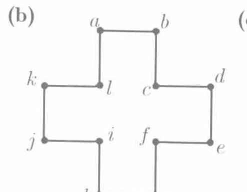
(b) 这些着色因旋转和反射而不同?

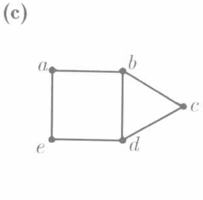
(c) 寻找这个六边形的两个 3 着色, 它们在 (a) 中是不同的, 而在 (b) 则是等价的.

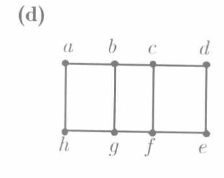
4. 3 种颜色的珠子可以形成多少种不同的 n 珠项链, 其中:
   (a) $n=7$; (b) $n=9$; (c) $n=10$; (d) $n=11$.

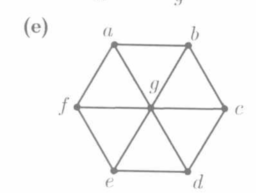
5. 求下面浮动图形的顶点的不同 m 着色数.

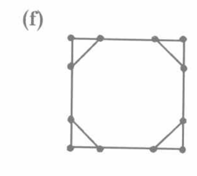
(a)


(b)


(c)


(d)


(e)


(f)


6. (a) 用边的置换, 而不是顶点的置换, 求正方形的对称群的循环指标.
   (b) 有多少种方法 3 着色浮动正方形的边?
   (c) 有多少种方法 3 着色浮动正方形的边和顶点?
   (d) 解释为什么下式不成立: (顶点的浮动 3 着色数)(边的浮动 3 着色数)=(边和顶点的浮动 3 着色数)?

7. 有多少种方法给练习 5 中的浮动图的边做 m 着色? (提示: 此时要考虑的是边对称的循环指数.)

8. (a) 当 (i)$n=3$, (ii)$n=4$, (iii)$n=7$ 时, 求每种颜色至少出现一次的不同的 3 着色 n 珠项链数.
   (b) 当项链可以反射和旋转时, 重做问题 (a).

9. 求不同的 2 面多米诺骨牌数 (这个多米诺骨牌的每一面都是空白或两个有 1 到 6 个点的正方形).

10. (a) 设 G 是 1, 2, 3, 4 的所有 4! 个置换的群. 求 P_G.
    (b) 使用 (a), 求用 3 种颜色之一给 4 个相同的弹球中的每一个着色的方法数 (把这个问题模拟成重复选择问题来检查你的答案).
    (c) 使用 (a), 寻找把从 3 种颜色中选出的 12 个球放入到 4 个相同的盒子里且每个盒子里有 3 个球的方法数.

11. (a) 求方格被着色成红色或黑色且在旋转下不同的 2×4 棋盘的数量.
    (b) (除旋转对称外) 互换棋盘方格上的红色和黑色得到另一个棋盘时, 认为这两个棋盘等价. 有多少不同的 2×4 棋盘?

12. 证明: 如果 $x_1^{k1}x_2^{k2}\cdots x_m^{km}$ 是凸 n 边形的对称 π 的循环指标 (用顶点的置换表示), 那么 $1k_1+2k_2+\cdots+mk_m=n$.

13. 在求解某个浮动图的顶点 2 着色数时, 假设我们知道这些 2 着色的置换产生的群 G^* 的循环指标 P_{G*}[如 9.1 节的 (1) 中那样], 而不知道这个图的对称群的循环指标 P_G. 为了得到 2 着色数, 可以用什么整数值取代 P_{G*} 的每一个 x_i? 还是说没有合适的替代? 请解释.

14. 设 S 是 n 个对象的集合, 而 G 是 S 的置换群. 对于 S 的子集 S_1, S_2, 定义等价关系 $S_1 \sim S_2$ 为对于某个 $\pi \in G$, 有 $S_1 = \pi(S_2)(= \{\pi(s)|s \in S_2\})$. 证明: 等价类数量等于 $P_G(2, 2, \cdots)$. (S 共有 2^n 个子集.)

15. 求凸 p 边形的顶点的 m 着色数, 其中 p 是素数 ($p > 2$), 如果:

(a) 只允许旋转;　　**(b)** 允许旋转和反射.

9.4 Polya 公 式

现在, 我们已经为描述终极目标, 模式目录公式, 做好了准备. 回想一下, 模式目录是一个生成函数, 它告诉我们使用一组不同的颜色可以给浮动图做多少着色. 对于浮动正方形的黑白着色而言, 模式目录是 $1b^4 + 1b^3w + 2b^2w^2 + 1bw^3 + 1w^4$. 例如, 项 $1b^3w$ 告诉我们有一个有 3 个黑色 (b) 顶点和 1 个白色 (w) 顶点的非等价着色. 模式目录中的系数可以看成是若干 Burnside 定理类型的计数问题的结果. 回想一下, Burnside 定理告诉我们, 由 S 的对称群产生的 S 的着色集合 T(其中 S 是正方形的顶点或四面体的面) 中的等价类中的数量 N 是:

$$N = \frac{1}{|G|}\sum_{\pi \in G} \Psi(\pi) \tag{$*$}$$

对于浮动正方形的 2 着色情况, 根据黑色顶点和白色顶点的数量, 我们把图 9.1 中的所有着色分成如下集合:

$$\begin{aligned}
T_0 &= \{C_1\}\\
T_1 &= \{C_2, C_3, C_4, C_5\}\\
T_2 &= \{C_6, C_7, C_8, C_9, C_{10}, C_{11}\}\\
T_3 &= \{C_{12}, C_{13}, C_{14}, C_{15}\}\\
T_4 &= \{C_{16}\}
\end{aligned}$$

如果设这个正方形的对称群只作用于集合 T_1 上, 那么 b^3w 的系数可以从 ($*$) 中得到. 回想一下, Burnside 定理的公式 ($*$) 可以运用于在 G 下封闭的任意着色集合. 短语 "在 G 下封闭" 的意思是指对于任意的 $\pi \in G$ 和任意的着色 $C \in T$, $\pi(C)$ 等于 T 中的另一个着色. T_1 是在 G 下是封闭的, 因为作用于有 3 个黑色顶点和 1 个白色顶点的着色上的任意对称产生另一个有 3 个黑色顶点和 1 个白色顶点的着色上的任意对称产生另一个有 3 个黑色顶点和 1 个白色顶点的着色. 对于其他的 T_k 也同样如此. 因此, 当 T_k 是 G 的作用集合时, $b^{4-k}w^k$ 在模式目录中的系数等于用 ($*$) 计数出的结果.

我们尝试着同时解决上面 5 个子问题. 在图 9.9 中我们复制了一份图 9.7 的表格, 而且又加上了新列 (iv). 在列 (iv) 的第一行, 我们写出一个多项式, 这个多项式的系数就是每个 T_k 中在 π_1 下保持不变的 2 着色数, 而在这一列的第二行, 我们写出一个多项式, 这个多项式的系数是每个 T_k 中在 π_2 下保持不变的 2 着色数, 然后

是在 π_3 下保持不变的 2 着色数, 以此类推. 于是我们把每一行中的项 b^4 加起来 (有 4 个黑色顶点的 2 着色数), 然后再除以 8, 得到 b^4 在模式目录中的系数, 再把每一行中的 b^3w 项加起来, 再除以 8, 得到 b^3w 的系数, 以此类推.

(i) 运动 π_i	(ii) 在 π_i 的作用下 不变的着色	(iii) 循环结 构表示	(iv) 在 π_i 的作用下不变的着色目录
π_1	16— 所有着色	x_1^4	$(b+w)^4 = 1b^4 + 4b^3w + 6b^2w^2 + 4bw^3 + 1w^4$
π_2	2—C_1, C_{16}	x_4	$(b^4+w^4) = 1b^4 + 1w^4$
π_3	2—$C_1, C_{10}, C_{11}, C_{16}$	x_2^2	$(b^2+w^2)^2 = 1b^4 + 2b^2w^2 + 1w^4$
π_4	2—C_1, C_{16}	x_4	$(b^4+w^4) = 1b^4 + 1w^4$
π_5	2—C_1, C_6, C_8, C_{16}	x_2^2	$(b^2+w^2)^2 = 1b^4 + 2b^2w^2 + 1w^4$
π_6	2—C_1, C_7, C_9, C_{16}	x_2^2	$(b^2+w^2)^2 = 1b^4 + 2b^2w^2 + 1w^4$
π_7	8—C_1, C_2, C_4, C_{10} $C_{11}, C_{12}, C_{14}, C_{16}$	$x_1^2x_2$	$(b+w)^2(b^2+w^2) = 1b^4 + 2b^3w + 2b^2w^2 + 2bw^3 + 1w^4$
π_8	8—C_1, C_3, C_5, C_{10} $C_{11}, C_{13}, C_{15}, C_{16}$	$x_1^2x_2$	$(b+w)^2(b^2+w^2) = 1b^4 + 2b^3w + 2b^2w^2 + 2bw^3 + 1w^4$

图 9.9

因为 π_1 的运动保持所有 C_s 不变, 所以第一行的系数是 1, 4, 6, 4, 1. 于是我们写出 $b^4+4b^3w+6b^2w^2+4bw^3+w^4$; 这就是不变着色的目录. 对于 π_1, 它的不变着色目录是所有着色的目录. 观察发现这个目录是 $(b+w)^4 = (b+w)(b+w)(b+w)(b+w)$, 每一个顶点对应一个 $(b+w)$. 对于 π_2, 它的不变着色目录是 b^4+w^4. 对于 π_3, 通过观察发现, 它的目录是 $b^4+2b^2w^2+w^4$. 这个表达式因式分解为 $(b^2+w^2)^2$.

正如我们之前在计数在某个 π 的作用下保持不变的着色总数时所做的那样, 我们来寻找这些不变着色的模式目录. 这一模式的关键仍然是, 在 π 的作用下保持不变的着色中, π 的循环内所有顶点必须有相同的颜色. 因为 π_2 有一个循环, 它包含所有 4 个顶点, 因此可能的着色是 4 个顶点全都是黑色或全都是白色; 因此, 它的目录是 b^4+w^4. 而运动 π_3 有两个 2 循环 (ac) 和 (bd). 在不变的着色中, 每个 2 循环使用两个黑色顶点或者两个白色顶点. 因此, 大小为 2 的循环的目录是 b^2+w^2. 有这样两个循环的所有运动有着色目录 $(b^2+w^2)(b^2+w^2)$.

π_i 的不变着色的目录是因子 (b^j+w^j) 的积, 其中, π_i 的每一个 j 循环对应一个因子. 所以, 我们需要知道 π_i 中各大小的循环数量. 但是, 这正是编码在循环结构表示中的信息. 实际上, 在结构表示中设置 $x_j=(b^j+w^j)$ 刚好产生 π_i 的不变着色目录. 根据这一方法, 我们计算不变着色目录的其余部分. 参见图 9.9. 具体地, 对于 π_7, 对照列 (ii) 中的着色列表, 查看它的目录. 把各不变着色的目录加到一起, 合并同类项, 再除以 8, 就可以得到模式目录.

正如之前在 9.3 节中那样, 我们得到一个更简洁的公式, 它因下面的步骤而节省了某些计算: 首先把循环结构表示加起来, 并除以 8, 然后设每一个 $x_j=(b_j+w_j)$, 并作多项式代数. 这种方法的第一步产生循环指标 $P_G(x_1, x_2, \cdots, x_k)$. 因此, 通过

设 P_G 中的 $x_j=b^j+w^j$, 我们得到模式目录.

如果允许使用 3 种颜色, 黑、白和红, 那么在不变着色中, 大小为 j 的每一个循环有目录 $(b^j+w^j+r^j)$. 所以, 可以设 P_G 中的 $x_j=(b^j+w^j+r^j)$. 上面的讨论适用于任意数量的颜色和任意的图形. 我们得到下面更具一般性的定理.

定理 (Polya 枚举公式) *设 S 是一个元素集合, G 是 S 的置换群, 这个置换群给 S 的着色建立一个等价关系. 使用两种颜色的 S 的非等价着色的目录由生成函数 $P_G((b+w),(b^2+w^2),(b^3+w^3),\cdots,(b^k+w^k))$ 给出. 使用颜色 $c_1,c_2,\cdots,c_m$ 的目录是:*

$$P_G\left(\sum_{j=1}^{m}c_j,\sum_{j=1}^{m}c_j^2,\cdots,\sum_{j=1}^{m}c_j^k\right)$$

我们暂时回到计数非等价 2 着色总数的问题. 这个数就是模式目录中各系数的和. 为了求系数和, 我们设不确定的 b 和 w(以及它们的幂) 等于 1, 或者等价地, 在 P_G 中设 $x_j=2$. 如果使用 m 种颜色, 我们在 P_G 中设 $x_j=m$, 从而得到与 9.3 节定理中相同的公式.

如许多其他类型的生成函数问题一样, 模式目录的生成函数的实际展开式可能是相当麻烦的. 我们要展开形如 $(c_1^i+c_2^i+\cdots c_m^i)^r$ 的表达式. 当 m 和 r 都变得很大时, 我们需要计算机代数软件的帮助.

例 1　3 珠项链的模式目录

确定使用黑珠子和白珠子且旋转下不同的 3 珠项链的模式目录. 使用黑、白和红珠子重复上面的问题.

根据 9.3 节的例 1, 我们知道 $P_G=\frac{1}{3}(x_1^3+2x_3)$. 用 $x_j=(b^j+w^j)$ 进行替换, 我们得到:

$$\begin{aligned}\frac{1}{3}[(b+w)^3+2(b^3+w^3)]&=\frac{1}{3}[(b^3+3b^2w+3bw^2+w^3)+(2b^3+2w^3)]\\&=\frac{1}{3}(3b^3+3b^2w+3bw^2+3w^3)\\&=b^3+b^2w+bw^2+w^3\end{aligned}$$

也可以根据经验得到上面的结果. 只有一种方法把所有珠子着色成全黑色或全白色. 如果一个珠子是白的, 而其他珠子是黑的, 那么通过旋转, 白珠子可以出现在任何地方; 因此, 有一个白珠子和两个黑珠子的项链只有一条. 根据对称性, 对于有一个黑珠和两个白珠子的项链也同样为真.

现在, 考虑使用 3 种颜色的 3 珠项链. 我们用 $x_j=(b^j+w^j+r^j)$ 在 P_G 中作替换, 得到 $\frac{1}{3}[(b+w+r)^3+2(b^3+w^3+r^3)]$. 我们不把这个表达式中的多项式展开, 而是使用间接的方法. 每一个目录项的系数可能还是 1. 有一个测试是否所有目录系数都等于 1 的一般方法: 把模式目录中的项数 N^* 与模式总数 N(即

$P_G(m,m,\cdots,m)$) 进行比较. 因为 N 等于目录中的系数和, 所以 $N^*=N$ 当且仅当每一项的系数是 1. 当 n 个元素用 m 种颜色着色时, 模式目录中的项数 N^* 等于 $C(n+m-1,n)$(参见练习 19). 对于我们的例子, $m=3,n=3$, 所以 $N^*=\binom{3+3-1}{3}=10$. 根据 9.3 节例 1, 我们知道 $N=11$. 因为 $N^*\neq N$, 所以我们知道不是所有项都有系数 1.

另一方面, 使 $N=N^*+1$ 的唯一方法就是目录中有 9 项系数是 1, 而有 1 项系数是 2. 但是, 如上所述, 只有一条所有珠子有相同颜色的项链, 而且也只有一条一个珠子的颜色是 A, 其他两个珠子的颜色是 B 的项链. 使用 3 种颜色的 3 珠项链的其他可能是每一种颜色有一个珠子. 因此, 必定有两条每种颜色有一个珠子的项链 (这两条项链是 3 颜色的任意循环顺序和反循环顺序), 所以黑、白和红 3 珠项链的模式目录是:

$$b^3+w^3+r^3+b^2w+b^2r+w^2b+w^2r+r^2b+r^2w+2bwr \qquad \square$$

例 2　7 珠项链的模式目录

求使用 3 个黑珠子和 4 个白珠子且在旋转下不同的 7 珠项链数.

我们需要确定 b^3w^4 在模式目录中的系数. 除 0° 旋转之外, 每一个旋转都是一个循环置换, 此时珠子数是素数 [参见练习 13(a)], 所以 $P_G=\frac{1}{7}(x_1^7+6x_7)$. 模式目录是 $\frac{1}{7}[(b+w)^7+6(b^7+w^7)]$.

因为模式目录中的因子 $6(b^7+w^7)$ 对 b^3w^4 的系数没有贡献, 所以可以忽视它. 因此 3 个黑珠子和 4 个白珠子的项链数是:

$$\frac{1}{7}[\text{ 在}(b+w)^7\text{中的}b^3w^4\text{的系数}]=\frac{1}{7}\binom{7}{3} \qquad \square$$

例 3　四面体边的 2 着色模式目录

求四面体的黑白边着色的模式目录.

尽管我们在 9.3 节的例 2 中计算过四面体的顶点对称循环指标, 但是还需要不同的边对称的循环指标. 因为着色的对象集合有 6 条边, 所以需要把这个四面体的对称考虑成边的置换.

显然, 0° 旋转保持所有边不变, 因此有循环结构表示 x_1^6. 120°(或者 240°) 是关于一个顶点和这个顶点的对面的中心线的旋转, 它循环地置换与这个顶点相连的边, 而且循环地置换这个顶点的对面的边界边 (参见 9.1 节的图 9.4). 因此它有循环结构表示 x_3^2.

180° 旋转是关于对边的旋转, 它保持这两条边不动 (参见图 9.4). 因为运用 180° 旋转两次就可以把这个四面体返回到原来的位置, 所以, 发生变化的其他 4 条边一定在 2 循环内. 因此, 180° 旋转有循环结构表示 $x_1^2x_2^2$, $P_G=\frac{1}{12}(x_1^6+8x_3^2+3x_1^2x_2^2)$.

用 $x_j=(b^j+w^j)$ 作替换, 我们得到:

$$\frac{1}{12}[(b+w)^6+8(b^3+w^3)^2+(b+w)^2(b^2+w^2)^2]$$
$$=\frac{1}{12}[(b^6+6b^5w+15b^4w^2+20b^3w^3+15b^2w^4+6bw^5+w^6)+(8b^6+16b^3w^3+8w^6)$$
$$+(3b^6+6b^5w+9b^4w^2+12b^3w^3+9b^2w^4+6bw^5+3w^6)]$$
$$=\frac{1}{12}(12b^6+12b^5w+24b^4w^2+48b^3w^3+24b^2w^4+12bw^5+12w^6)$$
$$=b^6+b^5w+2b^4w^2+4b^3w^3+2b^2w^4+bw^5+w^6$$

□

例 4　立方体顶点的 2 着色模式目录

求浮动立方体的顶点的 2 着色模式目录.

这个立方体的对称包括关于对面的旋转、对边的旋转和对顶点的旋转. 当然, 首先, 它有一个恒等对称, 其循环结构表示是 x_1^8.

(a) 对面: 作为具体的实例, 我们采用对面 $abcd$ 和 $efgh$ 的中心点连线的旋转; 参见图 9.10a. 首先, 90° 旋转产生置换 $(abcd)(efgh)$, 且有循环结构表示 x_4^2. 270° 旋转有相同的结构. 180° 旋转产生置换 $(ac)(bd)(eg)(fh)$, 所以循环结构表示是 x_2^4. 一共有 3 组对面, 所以这 3 组对面产生的对面旋转的循环指标是 $6x_4^2+3x_2^4$.

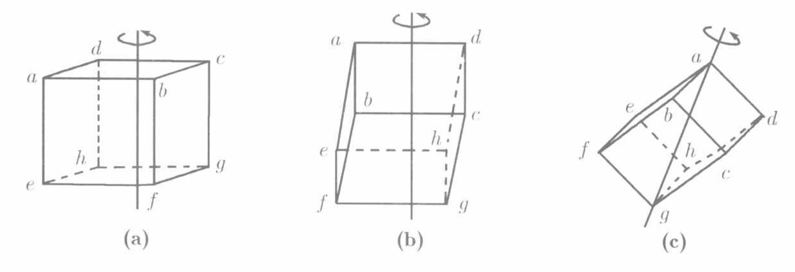


图 9.10　立方体的旋转: (a) 关于对面的旋转; (b) 关于对边的旋转; (c) 关于对顶点的旋转

(b) 对边: 作为具体实例, 我们做关于对边 ad 和 fg 中点连线的旋转; 参见图 9.10b. 180° 旋转产生置换 $(ad)(bh)(ce)(fg)$, 且循环结构表示是 x_2^4. 一共有 6 组对边, 所以它们产生对边旋转的循环指标是 $6x_2^4$.

(c) 对顶点: 作为具体实例, 我们做关于对顶点 a 和 g 的旋转; 参见图 9.10c. 120° 旋转产生置换 $(a)(bde)(chf)(g)$, 所以循环结构表示是 $x_1^2x_3^2$. 可以这样来观察这个置换都做了什么: 在保持 a 不动的任意运动的作用之下, 与 a 相邻的 3 个顶点 b,d,e 一定被循环地置换 (与 g 相邻的顶点也是类似). 240° 旋转与此有相同的结构. 一共有 4 对对顶点, 所以它们产生的对顶点旋转循环指标 $8x_1^2x_3^2$.

请读者自己验证我们已经枚举出了这个立方体的所有对称, 以及这些对称都不相同 (参见练习 17).

合并同类项, 我们求得 $P_G = \frac{1}{24}(x_1^8 + 6x_4^2 + 9x_2^4 + 8x_1^2x_3^2)$. 使用黑、白的立方体顶点着色的模式目录是:

$$\frac{1}{24}[(b+w)^8 + 6(b^4+w^4)^2 + 9(b^2+w^2)^4 + 8(b+w)^2(b^3+w^3)^2]$$

正如在前面的例子中那样, 已知项 b^8, b^7w, bw^7 的系数是 1. 而在上面所求的生成函数的 4 个因子中, b^6w^2, b^5w^3, b^4w^4 项分别是 $(\cdots + 28b^6w^2 + 56b^5w^3 + 70b^4w^4 + \cdots)$, $6(\cdots + 2b^4w^4 + \cdots)$, $9(\cdots + 4b^6w^2 + 6b^4w^4 + \cdots)$, $8(\cdots + b^6w^2 + 2b^5w^3 + 4b^4w^4 + \cdots)$. 求和, 然后再除以 24, 我们得到 $\frac{1}{24}(\cdots + 72b^6w^2 + 72b^5w^3 + 168b^4w^4 + \cdots) = 3b^6w^2 + 3b^5w^3 + 7b^4w^4$. (容易发现这些计算中的错误, 因为大多数错误会导致非整数系数.) 根据对称性, 我们要求的模式目录为:

$$b^8 + b^7w + 3b^6w^2 + 3b^5w^3 + 7b^4w^4 + 3b^3w^5 + 3b^2w^6 + bw^7 + w^8 \qquad \square$$

练 习 9.4

练习概况 前 16 道练习使用 Polya 枚举公式, 其余问题涉及与这一定理相关的一些理论. 注意, “浮动” 的意思是允许所有可能的旋转和反射.

1. 求浮动五边形的黑白顶点着色的模式目录.

2. 求黑白 n 珠项链的模式目录的表达式 (只旋转), 并求有 3 个白珠子, 其余是黑珠子的项链数, 其中 n 是:

　(a) $n = 6$; **(b)** $n = 9$; **(c)** $n = 10$; **(d)** $n = 11$.

3. 求浮动正方形顶点的黑、白和红着色的模式目录.

4. 求 4×4 棋盘中的 16 个方格的 2 着色模式目录的表达式 (只允许旋转).

5. 求 9.3 节练习 5 的浮动图的黑白顶点着色模式目录的表达式 (只允许旋转).

6. **(a)**求 (底为正方形的) 浮动金字塔的顶点 2 着色模式目录的表达式.

　(b) 对边着色, 重做问题 (a).

　(c) 对面着色, 重做问题 (a).

7. 对于 9.3 节练习 5 的浮动图, 求其边的 2 着色模式目录的表达式.

8. 求浮动立方体边的 2 着色模式目录的表达式, 并求有 3 条白边和 9 条黑边的边 2 着色数.

9. 求下列图形的面的 2 着色模式目录的表达式:

　(a) 浮动四面体; **(b)** 浮动立方体.

10. **(a)** 对于相邻顶点有不同颜色的浮动五边形, 求其顶点的 3 着色模式目录的表达式.

　(b) 对于浮动四面体, 重做问题 (a).

　(c) 对于浮动立方体, 重做问题 (a).

11. 求 4 个相同球的黑白着色模式目录的表达式 (参见 9.3 节练习 10).

12. 说明有 2 个白珠子和 $n-2$ 个黑珠子的 n 珠项链数是 $[n/2]$, 并对此给出粗略的解释 (不使用循环指标). ($[r]$ 是 $\leqslant r$ 的最大整数.)

13. 假定 p 是素数, 而且 p 珠项链是由黑珠子和白珠子组成的.

(a) 证明: 除 0° 旋转外, 每一个旋转都是顶点的循环置换.

(b) 正好有 k 个白珠子的项链数是多少?

14. (a) 假设我们不是对浮动正方形的顶点进行着色, 而是在其上放置 0, 1, 2 个相同的果冻. 求正方形各顶点上共有 k 个果冻的不同正方形的数量的生成函数, 并展开它. (提示: 在不变着色模式目录中, 因子内的指数是不同的.)

(b) 每个顶点得到一个红果冻、一个白果冻, 或一个红果冻和一个白果冻, 重做问题 (a). 现在, 求有 j 个白果冻和 k 个红果冻的正方形数量的生成函数.

15. 假设在 n 根接力棒的中央穿一个小洞, 然后用一根铁丝把它们串起来, 而且每根接力棒的两端都被涂上 3 种可能的颜色中的一种颜色 (红、白、蓝). 这根铁丝不动, 但是接力棒可以围绕其中心旋转. 有多少不同的形态, 当:

(a) $n = 2$;　　**(b)** $n = 3$.

16. 有多少不同的 (非同构的)4 顶点无标签图? (提示: 每条可能边有两个可能的颜色: "边出现" 和 "边不出现".)

17. 证明: 例 4 所列出的立方体的 24 个对称都是不同的, 且包含了立方体的所有对称.

18. 刻画这样一个几何图形, 当它浮动时只有一种这样的着色: 有一个顶点为黑色而其余顶点都是白色.

19. 证明: 凸 n 边形顶点的 m 着色模式目录中, 有 $C(n+m-1, n)$ 个不同项.

20. 设 $a_{k,n}$ 表示凸 n 边形顶点的所有 k 着色目录中系数的最少数目, 使得利用对称性我们可以知道这个模式目录中的其他所有系数. 求 $a_{k,n}$ 的生成函数 $g_k(x)$.

9.5　小结及参考文献

Polya 枚举公式很重要: 在实践方面, 它解决了许多重要的问题; 在数学方面, 它是群论和生成函数的巧妙结合; 在教学方面, 它是本书中你所看到的唯一强有力的组合公式 (通过把一些适当的数值 "插入" 到特殊的公式中, 我们可以解决很困难的问题). 探讨 Polya 公式的开发模式也同等重要: 9.3 节和 9.4 节可以看成是利用个案来进行数学理论的实验推导. (这种方法沿习了 Polya 在教授相关内容时所使用的教学模式.) 对这一理论的更严格的研究和它的扩展有兴趣的学生可以参考 [2], [2] 给出了 Polya 1937 年原始论文的英语翻译, 其论文题目是 "Kombinatorische Anzahlbestimmungen für Gruppen, Graphen und chemische Verbindungen".

在更一般化的研究中, 我们定义着色为从集合 S(顶点的集合) 到集合 R(颜色集合) 的函数. 对应于 C_6(见图 9.1) 的函数 f_6 可以写成表格形式 $\binom{a\ b\ c\ d}{w\ w\ b\ b}$. 我们定义一个运动和一个着色函数的合成 $\pi \cdot f$ 为一个新的着色函数 f'; 例如, $\pi_2 \cdot f_6$ 有映射 $a \to b \to w, b \to c \to b, c \to d \to b, d \to a \to w$, 或 $\pi_2 \cdot f_6 = \binom{a\ b\ c\ d}{w\ b\ w\ b} = f_7$. 由对称 π 所引发的着色置换把每个 f 映射到 $\pi \cdot f$. 尽管很麻烦, 但是这样一来, (在通过观察进行计算之前) 我们就有可能形式地计算着色置换. 我们定义着色等价为: $f \sim f'$ 当且仅当存在 π 使得 $\pi \cdot f = f'$. 可以用这些新定义重述我们的理论.

Polya 枚举公式在组合数学的另一个领域有重要的应用. 我们通常使用它来枚举图 (参见 9.4 节练习 16). F. Harary 是这一应用的先驱. 参见 Harary and Palmer[1].

1. F. Harary and E. Palmer, *Graphical Enumeration*, Academic Press, New York, 1973.

2. G. Polya and R. C. Reade, *Combinatorial Enumeration of Groups, Graphs, and Chemical Compounds*, Springer-Verlag, New York, 1987.

第 10 章　计算机科学在枚举中的应用

10.1　生成排列和组合, 程序设计项目

本章从计算机科学的角度看枚举. 这里给出 3 种方法: 首先, 构建以某种“自然顺序”列举一个集合中的结果的算法; 第二, 使用形式语言来枚举结果集合; 第三, 使用有限状态机识别给定集合中的结果.

我们在第 5 章使用整整一章讲述了不同类型的排列与选取, 现在, 我们应该说一说枚举, 或列举出这些排列和组合. 本节给出枚举 n 元集合的所有排列以及 n 元集合的所有 r 组合的算法. 运用这些算法, 我们可以编写枚举各组合集合中所有元素的程序; 我们将在本节的后半部分做这件事.

下面这些例子说明如何在运筹学中使用这样的算法. 假设要在一个月内在一家工厂生产 8 种不同的产品. 给定从产品 i 到产品 j 的转换成本, 我们必须寻找使产品转换成本之和最小的产品生产顺序. 这个问题的最简单的解决方案是生成产品的所有排列, 并检查其中哪一种顺序的成本最少 (3.3 节给出了解决这类问题的“分支定界”方法).

假设 10 名工人中的每一名都熟知某个项目所需的 30 种技术中的若干不同技术, 而我们必须寻找所有这样的 5 人组, 即, 这 5 个人联合起来熟知所有 30 项技术. 为了寻找满足这一特殊条件的子集, 我们需要检查 10 名工人的所有 5 人子集 (也可以使用“分支定界”方法来解决这一问题).

枚举排列和组合有若干不同的算法. 我们将介绍以**词典顺序**(lexicographic order) 枚举这些集合的算法, 即, 我们把对象的排列处理成构成“单词”的“字母”序列, 并以字母表顺序列出这些“单词”. 这一方法要求必须对对象编号 (或排序). 我们写 $A < B$, 如果按词典顺序序列 A 小于 B. 类似地处理子集, 即用字母递增顺序排列每一个子集中的“字母”(对象). 例如, 表 10.1 按词典顺序列出 a, b, c 的所有排列, 而表 10.2 则按词典顺序列出 1, 2, 3, 4, 5 的所有 3 组合.

表　10.1

1.	abc
2.	acb
3.	bac
4.	bca
5.	cab
6.	cba

表　10.2

1.	123	6.	145
2.	124	7.	234
3.	125	8.	235
4.	134	9.	245
5.	135	10.	345

为了以词典顺序列出 n 元集合的所有排列，我们需要能够确定：对于任意给定的排列 $A = a_1a_2a_3\cdots a_n$，按词典顺序，它的下一个排列 $B = b_1b_2b_3\cdots b_n$ 是什么. 如果 $a_{n-1} < a_n$，那么交换 a_{n-1} 和 a_n 在 A 中的位置将产生词典顺序下的更大序列，这就是所求的序列 B. 例如，14325 的后面跟着的是 14352.

假设 $a_{n-1} > a_n$. 那么交换 a_{n-1} 和 a_n 在 A 中的位置将产生词典顺序下更小的序列. 我们必须转而处理 a_{n-2}. 如果 $a_{n-2} < a_{n-1}$，那么可以重新排列 A 的最后 3 个数字来产生词典顺序下更大的序列. 为了尽可能小地增加这个序列，位置 $n-2$ 应该有从 a_{n-1} 和 a_{n-2} 中选出的下一个更大的"数字"(大于 a_{n-2}). 原来最后 3 位数字中剩余的两个数字以递增顺序进入位置 $n-1$ 和 n. 例如，14352 后面应该跟着 14523，而 14523 后面应该跟着 14325. 如果 $a_{n-2} > a_{n-1}$(且 $a_{n-1} > a_n$)，那么就必须转而处理 a_{n-3}.

一般地，我们从序列的右端开始向左移动，查看第一对使 $a_i < a_{i+1}$ 的 a_i, a_{i+1}(且 $a_{i+1} > a_{i+2} > a_{i+3} > \cdots > a_{n-1} > a_n$). 这时，通过在位置 i 处放置 $a_k=\min\{a_j|a_j > a_i$ 且 $j > i\}$并在位置 $i+1$ 到 n 以递增顺序放置其余的数字 $a_i, a_{i+1}, \cdots, a_n(a_k$ 除外)，我们可以得到下一个更大的序列.

以词典顺序列出排列的下一个序列的算法

```
PROCEDURE NEXTPERMUT(A, n)
Comment—find lexicographically next permutation of n-element array A
Syntactic Comment—GOTO's are used instead of WHILE
statements for simplicity.
BEGIN
    FOR i ← n−1 TO 1 STEP −1 DO
      IF A(i)<A(i+1) THEN GOTO 1);
      Comment —find min A(h) such that A(h)>A(i) and swap A(h), A(i)
1) FOR h ← n TO i+1 STEP −1 DO
       IF A(i)<A(h) THEN GOTO 2);
2) TEMP ← A(i); A(i)← A(h); A(h) ← TEMP;
    Comment—complete next arrangement by reversing rest of sequence
    FOR h ← i+1 TO n DO TME(h) ← A(h);
    FOR h ← i+1 TO n DO A(h) ← TEM(n+j+1−h);
END NEXTPERMUT.
```

接下来说明如何以词典顺序生成 n 元集合的所有 r 组合. 为了简单起见，假设 n 元集合是整数 $1, 2, \cdots, n$ 的集合. 回想一下，我们以递增顺序列出每一个 r 组合中的元素 (参见表 10.2). 我们从右到左搜索直到发现第一个 $a_i \neq n-r+i$ 的元素来寻找当前 r 组合 $a_1a_2\cdots a_r$ 的后继者 $b_1b_2\cdots b_r$. 对于这个 i，设 $b_i = a_i+1, b_j = b_{j-1}+1,$

其中 $i < j \leqslant r$.

以词典顺序列出组合的下一个子集的算法

```
  PROCEDURE NEXTSUBSET (A, n, r)
  Comment—given an r-subset A, find the
  lexicographically next r-subset from 1, 2, 3, ···, n.
BEGIN
    FOR i ← r TO 1 STEP -1 DO
       IF A(i) < n-r+i THEN GOTO 1);
1) A(i) ← A(i)+1;
    FOR h ← i+1 TO r DO A(h) ← A(h-1)+1;
END   NEXTSUBSET.
```

程序设计项目

使用前面的算法, 可以枚举大多数组合集合. 回想一下 5.3 节, 我们可以把重复选取问题建模成无重复选取问题, 并把重复排列问题建模成一系列选取问题.

下面几个例子给出了用于列出前面章节的两组枚举问题中所有结果的计算机程序.

例 1　枚举所有配对

列出一次舞会上把 10 名女子与 20 名男子中的 10 名男子配对的所有方法.

设男子被标号为 $1, 2, 3, \cdots, 20$. 设 B 是这样的一个数组, 它表示出了与 10 名女子配对的当前 10 名男子的标号 (有 10! 种方式). 设 $A(i)$ 是与第 i 个女子配对的男子, $i = 1, 2, \ldots, 10$. NEXTPERMUT和NEXTSUBSET是本节早前给出的过程 (算法).

```
BEGIN
   Comment—construct lexicographically first subset of 10 men
      FOR i ← 1 TO 10 DO B(i) ← i;
   Comment—loop to list all 10-subsets of the 20 men
      FOR i ← 1 TO (20 choose 10) DO
         BEGIN
            FOR i ← 1 TO 10 DO A(i) ← B(i);
   Comment—loop to list all arrangements of men in
   subset A with the women
      FOR k ← 1 TO 10!  DO
         BEGIN
            print out pairing (i, A(i), i=1, 2,···, 10);
            NEXTPERMUT (A, 10)
         END;
```

```
    NEXTSUBSET (B, 20, 10);
      END;
END.
```

□

例 2 枚举没有连续 I 的排列

列出VISITING的字母排列中没有成对 I 连续出现的所有排列.

设OUTCOME(k)(k=1, 2, $\cdots$, 8) 是用于构建可能结果的数组. 设CONS(k)(k=1, 2, $\cdots$, 5) 是OUTCOME中所使用的辅音 V, S, T, N, G 的置换.

设POS(j)(j=1, 2, 3) 是在数组OUTCOME中 I 将出现的位置. 考虑 VISITING 中的 5 个辅音 G, N, S, T, V 的前面、中间和后面这 6 个位置. 我们挑出这 6 个位置中的 3 个位置的子集来放置 I(这是 5.4 节练习 51 所讨论的没有连续元素的方法). 假设挑出位置 2, 3, 6. 位置 2 上的第一个 I 是在第一个辅音和第二个辅音之间, 所以第一个 I 进入位置 2. 位置 3 上的第二个 I 是在第二个辅音和第三个辅音之间, 而且在第一个 I 之后, 所以第二个 I 应该在位置 4(而不是位置 3) 上. 位置 6 上的第三个 I 出现在所有 5 个辅音之后, 而且出现在其他两个 I 之后, 所以第三个 I 进入位置 8. 一般地, 第一个 I 的位置是这个 3 元子集的第一个数. 第二个 I 的位置是这个 3 元子集的第二个数加 1. 第三个 I 的位置是这个子集中的第三个数加 2.

```
BEGIN
   Comment—construct lexicographically first 3-subset
   of 1, 2, 3, 4, 5, 6
      FoR i ← 1 TO 3 DO P(i) ← i;
   Comment—loop on all 3-subsets of 1, 2, 3, 4, 5, 6.
      FOR i ← 1 TO (6 3) DO
         BEGIN
            Comment—convert 3-subset into positions
            for Is:
               FOR k ← 1 TO 3 DO POS(k) ← P(k)+k-1;
            Comment—construct lexicographically first
            permutation of consonants
               CONS(1) ← 'G'; CONS (2) ← 'N';
               CONS(3) ← 'S'; CONS (4) ← 'T';
               CONS(5) ← 'V';
            Comment—loop on all permutations of the
            five consonants
                 FOR j ← 1 TO 5!  DO
                     BEGIN
            Comment—construct current OUTCOME
```

```
                    i1 ← 1; j1 ← 1;
                    FOR k ← 1 TO 8 DO
                    IF k=POS(i1)  THEN
                        BEGIN
                            OUTCOME(k) ← '1'; i1 ← j1+1
                        END;
                     ELSE
                        BEGIN
                            OUTCOME(k) ← CONS (j1); j1 ← j1+1;
                        END;
                     PRINT OUTCOME (i), i=1, 2, ···, 8;
                     NEXTPERMUT (CONS, 5);
               END;
           NEXTSUBSET (P, 6, 3);
     END;
END.
```

□

练 习 10.1

练习概况 前 5 道练习使用本节所给出的 3 个算法. 接下来的 6 道练习涉及这些算法的变形和相关性质. 最后 9 道练习是程序设计项目.

1. 按词典顺序枚举 1, 2, 3, 4 的所有排列.

2. 按词典顺序枚举 a, c, e, h 的所有排列.

3. 按词典顺序枚举 1, 2, 3, 4, 5, 6 的所有 4 元子集.

4. 按词典顺序枚举 a, d, h, j 的所有 3 元子集.

5. 在下面的集合中, 词典顺序下的第一个元素和最后一个元素是什么:

(a) 整数 $1, 2, \cdots, 10$ 的所有排列的集合?

(b) 26 个字母的所有 4 元子集的集合?

6. 用归纳法证明本节所给出的后继者法则可以正确地寻找下列集合中的在词典顺序下的下一个元素:

(a) n 元集合的所有排列的集合.

(b) n 元集合的所有 r 元子集的集合.

7. 设计另一个枚举 n 元集合的所有排列的算法.

8. 设计另一个枚举 n 元集合的所有 r 元子集的算法.

9. (a) 开发一个过程来计算 $1, 2, \cdots, n$ 的给定排列在其所有排列的词典顺序下的位置.

(b) 排列 314526 在 $1, 2, \cdots, 6$ 的所有排列的词典顺序下的位置是什么?

10. (a) 开发一个过程来计算 $1, 2, \cdots, n$ 的给定 r 元子集在所有 r 元子集的词典顺序下的位置.

(b) 子集{3, 4, 7}在 1, 2, ⋯, 8 的 3 元子集的词典顺序下的位置是什么?

11. 设计一种以词典顺序列出 n 元集合的所有子集的方法.

12. 下表给出了从工作 i 转换到工作 j 的成本. 为了最小化转换成本, 应该以什么顺序执行 4 种工作?

从 \ 到	1	2	3	4
1	—	4	5	2
2	6	—	4	8
3	3	5	—	7
4	2	7	6	—

13. 下表告诉我们哪些人了解交易集合中的不同交易. 求联合起来了解所有 10 种交易的所有 4 人子集.

人员 \ 贸易	1	2	3	4	5	6	7	8	9	10
1	1	1	0	1	0	0	0	1	0	0
2	1	0	1	0	1	0	1	0	0	1
3	0	1	0	0	0	1	0	0	1	1
4	0	0	1	1	1	0	1	1	0	0
5	0	1	0	1	0	0	1	0	1	0
6	1	0	1	0	1	0	0	0	1	0
7	1	0	0	1	0	1	0	0	0	1

14. 一名旅行商要遍历 5 个城市进行一次推销旅行, 开始并结束于城市 1. 使用下面的城市间的成本矩阵求最廉价的推销路线.

从 \ 到	1	2	3	4	5
1	—	4	5	3	6
2	7	—	3	4	7
3	3	4	—	3	6
4	2	5	6	—	5
5	3	3	7	4	—

15. 编写一个列出 1, 2, 3, 4, 5, 6 的所有重复 5 元子集的程序.

16. 编写一个列出 3 个 a、3 个 b 和 3 个 c 的所有排列的程序.

17. 把 r 个相同的对象分配到 n 个不同盒子且每个盒子有 3 到 6 个对象, 编写一个列出所有分配的程序.

18. 编写一个列出 $x_1 + x_2 + \cdots + x_r = n$ 的所有整数解的程序:

(a) $x_i \geqslant k$ (b) $0 \leqslant x_1 \leqslant x_2 \leqslant x_3 \leqslant \cdots \leqslant x_r$

19. 把 r 个不同的对象分配到 n 个不同的盒子且每个盒子至少有 m 个对象, 编写一个列出所有分配的程序.

20. 为列出 $1, 2, \cdots, n$ 的没有成对连续整数的所有 r 元子集编写一个程序.

10.2 形式语言和文法

计算机科学使用形式语言来研究程序设计语言. 我们将使用形式语言生成计

数问题的所有结果集合. 第 5 章的计数公式通常是以对结果的不同成分的分析为基础的. 例如, 在计数一手 5 张牌中一种点数有 4 张的手牌数时, 我们是通过把生成一种点数的 4 张牌的方法数乘以挑出第 5 张牌的方法数而得到结果的. 通过严格描述如何生成所有完整结果的列表, 我们对第 5 章所讨论的组合推理有一种崭新的看法.

文法 $G =< N, T, P, s >$ 的组成是:

(1) 非终结符的有穷集合 N;

(2) 终结符的有穷集合 T;

(3) **产生式规则**(production rule) 的有穷集合 P, 其成员都有 $\alpha \to \beta$ 的形式, 其中 α 和 β 是集合 $N \cup T$ 中的元素的串;

(4) 一个特殊的开始符号 $s \in N$.

文法为生成关联于这个文法的每一个正确 "句子" 提供了所需的规则和开始符号. 这种文法结构的背景是程序设计语言和自然语言的设计. 对于自然语言, 文法的句子是标准的句子, 终结符是这一语言的单词, 而非终结符则是话语的部分和句子的成分, 例如, 形容词和主语. 对于我们的组合目标来说, 句子是某个组合集合中的可能结果, 诸如有两个对子的一手 5 张牌, 而终结符是构成结果的对象或符号, 诸如不同的纸牌.

我们说语句 $\gamma\alpha\delta$ **直接推导出**(directly derive)$\gamma\beta\delta$, 写作 $\gamma\alpha\delta \Longrightarrow \gamma\beta\delta$, 如果 $\alpha \to \beta$ 是 G 的产生式, 而 γ 和 δ 是 $N \cup T$ 的符号串. 我们说 ω_1 推导出 ω_n, 写成 $\omega_1 \to \omega_n$, 如果存在直接推导的序列 $\omega_1 \Longrightarrow \omega_2 \cdots \Longrightarrow \omega_n$. G 的**句子**(sentence) 是从符号 s 推导出的终结符序列 ω. 由 G 推导出的语言 $L(G)$ 是 G 的所有句子的集合.

例 1 没有连续 1 的二进制序列

构造语法 G, 使得它的语言 $L(G)$ 是由没有连续 0 的所有二进制序列组成的.

设终结符号集合是{0, 1}. 开始符号 s 一定在非终结符集合中. 我们根据需要引入其他的非终结符. 如果要生成所有二进制序列 (包括空序列 ϕ), 那么产生式规则应该是 $s \to 1s$, $s \to 0s$, $s \to \phi$. s 出现在前两个产生式规则两端的递归形式是计算机程序设计中的标准技术. 为了避免两个 0 连续出现, 我们需要用迫使任意的 0 后面都跟着 1(或 0 在语句末尾) 的规则取代第二个产生式规则. 一个这样的产生式规则集合是:

$$G_1 : s \to 1s, \quad s \to 01s, \quad s \to 0, \quad s \to \varnothing$$

另一个可能集合是 (开始符号为 s'):

$$G_2 : s' \to 1s', \quad s' \to 0t', \quad t' \to 1s', \quad s' \to 1, \quad s' \to 0, \quad t' \to 1$$

例如, 语句 01101 在 G_1 中可以这样推导出来:

$$s \Rightarrow 01s \Rightarrow 011s \Rightarrow 01101s \Rightarrow 01101$$

而在 G_2 中可以这样推导出来:

$$s' \Rightarrow 0t' \Rightarrow 01s' \Rightarrow 011s' \Rightarrow 0110t' \Rightarrow 01101$$

观察 G_1 的前两个产生式规则, 它们很自然地蕴含着不含连续 0 的 n 位二进制序列数目 a_n 的递归关系. 这一递推关系是:

$$a_n = a_{n-1} + a_{n-2}$$

例 1 的文法中的递归形式是现代程序设计语言文法的精髓, 这样的文法以开始符号 < 语句 > 开始进行递推, 例如,

$$< \text{语句} > \rightarrow < \text{IF语句} >$$

$$< \text{IF语句} > \rightarrow \text{IF} < \text{条件} > \text{THEN} < \text{语句} >$$ □

例 2 第一个 0 在第一个 1 之前的三进制序列

构造一个文法 G, 其语言 $L(G)$ 是由第一个 0 出现在第一个 1 之前的所有三进制序列组成的; 可以没有 1 出现, 但是至少出现一个 0. 同时, 给出 G_n, 其语言 $L(G_n)$ 是由长度为 n 的这样的序列组成的.

所有这样的序列所要求的文法必须生成这样的句子: 以若干 2 开始的子序列, 后面是一个 0. 在 0 出现之后, 对于其余位置上的数字再没有任何限制.

$$G_3 : s \rightarrow 2s, \quad s \rightarrow 0t, \quad t \rightarrow 0t, \quad t \rightarrow 1t, \quad t \rightarrow 2t, \quad t \rightarrow \varnothing$$

例如, 句子 22021 可以这样推导出来:

$$s \Rightarrow 2s \Rightarrow 22s \Rightarrow 220t \Rightarrow 2202t \Rightarrow 22021t \Rightarrow 22021$$

根据这个文法, 显然在第一个 1 之前出现 0 的所有 n 位三进制序列数是:

$$\sum_{k=0}^{n-1} 3^{n-k-1}$$

其中, k 代表第一个 0 前面的 2 的数量 (这个 0 之后的剩余 $n-k-1$ 位数字可以任意挑选).

现在考虑如何修改前面的文法使得它的所有句子都有给定长度 n. 我们必须把文法 G_n 构造成 "计数 n". 假设 $n \geqslant 2$. 为了做到这一点, 我们需要一组带下标的非终结符号 s_i, t_i, 其中 $i = 2, \cdots, n$,

$$G_n : s \rightarrow 2s_2, \quad s \rightarrow 0t_2$$

$$s_i \rightarrow 2s_{i+1}, \quad s_i \rightarrow 0t_{i+1}, \quad i = 2, \cdots, n-1$$

$$t_i \to 0t_{i+1}, \quad t_i \to 1t_{i+1}, \quad t_i \to 2t_{i+1}, \quad i = 2, \cdots, n-1$$

$$s_n \to 0, \quad t_n \to 0, \quad t_n \to 1, \quad t_n \to 2$$

□

在生成枚举问题所有结果的意义之下, 像 G_1 和 G_3 这样的简单文法与生成函数很类似. 即, 它们生成各种大小的结果, 而不是只生成给定长度的结果. 注意, 对于有限枚举问题, 我们总是可以构建一个庞大的文法, 使得每一个可能的结果都是某个产生式的右端项 (而 s 在左边).

例 3 满堂红的一手 5 张牌

构造一个文法 G, 其语言 $L(G)$ 是所有满堂红的 5 张手牌 (满堂红是一种点数有 3 张牌, 另一种点数有 2 张牌的 5 张手牌).

这一文法相当容易构建, 但是它需要很多终结符和非终结符. 这样的一手牌可以分解成两个部分, 一种点数的 3 张和一个对子. 我们先生成一种点数的一手 3 张牌. 第一种类型的产生式规则是:

$$\text{对所有可能的 } i \neq j, i, \quad j = 1, 2, \cdots, 13, \quad s \to t_i p_j$$

其中, 下标的范围是 13 种点数. 接下来, 我们必须使用第 i 种点数, 从每个 t_i 推导出第 i 种点数的所有 3 张牌, 并从每个 p_j 推导出第 j 种点数的所有两张牌. 最简单的方法是对于每个可能同点数的 3 张牌都有一个产生式规则:

$$t_i \to c_{is}c_{ih}c_{id}, \quad t_i \to c_{is}c_{ih}c_{ic}, \quad t_i \to c_{is}c_{id}c_{ic}, \quad t_i \to c_{ih}c_{id}c_{ic}$$

其中, c_{ix} 是花色 x(黑桃、红心、方片、梅花) 的第 i 种点数的纸牌. 对子也类似.

满堂红的 5 张牌数量公式是 13×12×(一种点数 3 张牌的数量)×(一种点数一对牌的数量)=$13 \times 12 \times C(4, 3) \times C(4, 2)$. □

作为一般原则, 形式文法对生成有很多类型的终结符的 (无序) 子集或组合不是很有效. 而且, 正如上面所提到的那样, 构建生成给定长度的序列的文法很麻烦. 而另一方面, 文法很容易生成很多组合集合. 而且, 当找到一个组合集合的文法时, 对于求给定长度的结果数量, 它会指明一种寻找递推关系或组合公式的方法.

有很多特殊类型的文法, 它们的产生式规则有特定的形式. 关于文法类型的更好研究, 参见 Hopcroft, Motwani, and Ullman[1]. 这里, 我们只讨论一种特殊类型的文法, 称为**正则文法**(regular grammar). 由正则文法生成的语言称为**正则语言**(regular language). 在正则文法下, 所有产生式规则都有下面的形式:

$$t \to at' \quad \text{或} \quad t \to a, \quad t, t' \in N, \quad a \in T(\text{有可能} t = t')$$

例 1 中的文法 G_2 和例 2 中的文法 G_n 都是正则的. 注意, 尽管 G_2 不是正则的, 但是例 1 中的文法 G_1 与 G_2 等价. 等价的 (equivalent) 语法产生相同的语言. 仔细观察 G_2 是如何消除 G_1 中的非正则产生式 $s \to 01s$ 和 $s \to \phi$ 的.

G_3 能够转化成一个等价的正则文法 G' 吗? 答案是肯定的. G_3 中唯一的非正则产生式是 $t \to \phi$. 这一规则是用于 G_3 的任意句子推导的最后产生式. 为了取代 $t \to \phi$, 我们必须在推导中往回退一步. 参见例 2 中句子 22021 的推导例子. 22021 所使用的倒数第二个产生式是 $t \to 1t$. 新规则 $t \to 1$ 与 $t \Rightarrow 1t \Rightarrow 1$ 等效. 我们必须为右边出现 t 的每一个产生式实例构建类似的新规则: $t \to 0t, t \to 2t, s \to 0t$. 所求的新文法是:

$$
\begin{aligned}
G_3' : s \to 2s, \quad s \to 0t, \quad s \to 0,\\
t \to 0t, \quad t \to 1t, \quad t \to 2t\\
t \to 0, \quad t \to 1, \quad t \to 2
\end{aligned}
$$

10.3 节将研究正则文法和语言的一个非常特殊的性质, 因此, 知道由给定的文法生成的语言是否有等价正则文法很重要. 下面是一个没有正则文法的简单语言的标准例子. □

例 4 没有正则文法的语言

构建一个生成语言 L 的文法, 其中 L 的句子是所有的 n 个 0 后面跟着 n 个 1 的二进制序列. 证明这样的语言不存在正则文法.

为这种语言构建适当的文法的关键是从中间向两边推导出序列.

$$G_4 : s \to 0s1, \quad s \to 01$$

根据正则文法的要求, 从左到右生成这样序列会遭遇这样的问题: 必须在开始生成 1 时, 使用不同的非终结符来记住它生成了多少个 0. 通过简单的思考, 下面类似于 G_n 的文法似乎是正确的:

$$G_5 : s \to 0s_1, \quad s_i \to 0s_{i+1}, \quad s_i \to 0t_{i+1}, \quad t_i \to 1t_{i+1}, \quad s_i \to 1, \quad t_i \to 1$$

因为我们对 n 的大小没有限制, 下标 i 必须有无限的范围, 即使产生式的数量是有限的. 然而, 文法只能有有限的元素和产生式. □

练 习 10.2

1. 给出下列句子的推导.
   (a) 例 1 的 G_1 中的 101101.
   (b) 例 2 的 G_3 中的 220102.
   (c) 例 2 的 G_n 中的 20202, 其中 $n = 5$.
   (d) 例 3 中的 $c_{3s}c_{3h}c_{3c}c_{8h}c_{8d}$.
   (e) 例 4 的 G_4 中的 00001111.
2. 设文法为 $s \to tt, t \to 01t, t \to 10t, t \to 0, t \to 1$, 给出下面各句子的推导.
   (a) 0010.
   (b) 1001.
   (c) 001100.
3. 对于练习 2 文法中的句子 010101, 给出 3 个推导.

4. 构建能够生成以 1 开始且以 0 结束的所有二进制序列的文法.

5. 为了生成若干 0 的后面跟着若干 1 的所有二进制序列 (0 的数量不必等于 1 的数量), 构建相应的文法.

6. 构建能够生成所有 5 位二进制序列的文法.

7. **(a)** 构建能够生成 Mastermind 游戏中的所有可能密码的文法 (参见绪论).

(b) 对于正好有一个 R 的密码, 重做问题 (a).

(c) 对于至少有一个 R 的密码, 重做问题 (a).

8. 构建能够生成满足下面条件的所有 5 张手牌的文法.

(a) 有一种点数的 4 张牌. **(b)** 有一个顺子.

9. **(a)** 从每堆由 2 个 A、2 个 B 和 2 个 C 的若干堆对象中重复选出 4 个对象，构建能够生成其所有子集的文法.

(b) 当每个字母至少有一个时, 重做问题 (a).

10. 把练习 2 中的文法转换成一个正则文法.

11. 一枚硬币投掷 5 次, 其中第二个正面正好出现在最后一次, 为生成这样的投掷序列的所有结果构建正则文法.

12. 构建能够生成正好有两个 1 的所有 6 位二进制序列的正则文法.

10.3　有限状态机

本节给出一种最简单、理想化的计算机, 称为有限状态机. **有限状态机**$M = \langle I, O, S, f, \theta\rangle$ 的组成是:

(1) 输入元素的有穷集合 I;

(2) 输出元素的有穷集合 O;

(3) 内部状态的有穷集合 S;

(4) 下一个状态函数 $f: I \times S \to S$;

(5) 输出函数 $\theta: I \times S \to O$.

如果一台机器从某个初始状态 s_1 开始, 并接受 I 的元素构成的输入序列 $\alpha = a_1a_2\cdots a_n$, 那么读入第一个输入元素 a_1 后, 这台机器打印输出元素 $\theta(a_1, s_1)$, 并切换到内部状态 $f(a_1, s_1)$. 随后的输入元素有类似的效应. 如果机器在状态 s_i 读入 a_i, 那么打印元素 $\theta(a_i, s_i)$, 机器切换到状态 $f(a_i,\ s_i)$. 处理完最后一个输入元素之后, 机器停止.

例 1　计数 1 的串数

考虑有限状态机 M, 其中, $I = \{0,1\}, O = \{0,1,2,3\}, S = \{t_0, s_1, t_1, s_2, t_2, s_3\}$, 且 f 和 θ 由表 10.3 给出 (表中每一项的左边元素是内部状态, 右边元素是输出). 我们说 M 计数 (并打印) 至此为止的输入序列中的 1 的串数, 最多计数 3 个串. 回想一下, 1 的串是 1 的连续集合 (前面和后面都是 0). 读入最后的输入元素之后, 最后的输出打印的是整个输入序列中的 1 的串数. 例如, 假设 M 从状态 t_0 开始的

话, 输入序列 0110010 将产生下面的内部状态和输出的序列:

表 10.3

	0	1
t_0	$t_0/0$	$s_1/1$
s_1	$t_1/1$	$s_1/1$
t_1	$t_1/1$	$s_2/2$
s_2	$t_2/2$	$s_2/2$
t_2	$t_2/2$	$s_3/3$
s_3	$t_3/3$	$s_3/3$

$$(t_0,-)\xrightarrow{0}(t_0,0)\xrightarrow{1}(s_1,1)\xrightarrow{1}(s_1,1)\xrightarrow{0}(t_1,1)\xrightarrow{0}(t_1,1)\xrightarrow{1}(s_2,2)\xrightarrow{0}(t_2,2)$$ □

例 2 没有连续 1 的二进制序列

考虑有限状态机 M, 其中, $I=\{0,1\}, 0=\{Y,N\}, S=\{s_0,s_1,s_2\}$, f 和 θ 由表 10.4 给出. 机器从状态 s_0 开始. 这台机器打印 Y, 直到它发现两个连续的 0 为止. 如果两个 0 在一起出现, 那么它之后永远打印 N. 因此, 如果输入序列不包含成对的连续 0, 则打印出的最后输出元素是 Y(表示 “是”), 否则最后打印的是 N(表示 “不是”). □

表 10.4

	0	1
s_0	s_1/Y	s_0/Y
s_1	s_2/N	s_0/Y
s_2	s_2/N	s_2/N

我们说例 2 中的机器*识别*(recognize) 没有连续 0 的二进制序列的集合. 用 10.2 节的话说, 这台机器识别 10.2 节例 1 中的文法 G_1 和 G_2 所产生的语言. 我们将在本节的后半部分形式地定义识别.

例 3 词典顺序下的下一个子集

设计一台有限状态机, 当输入序列是{1, 2, 3, 4, 5, 6}的 3 元子集时, 它将打印出词典顺序下的下一个 3 元子集.

回想一下, 如果我们把数字的一个子集以数字的递增顺序写成一个数, 于是词典顺序下的下一个子集就是这样的子集, 它的数字形成下一个较大的数. 例如, 在我们的 3 元子集中, 在词典顺序下, 235 后面跟着 236, 而 236 后面跟着 245. 10.1 节给出求词典顺序下 n 元集合的下一个 r 元子集的算法: 求当前子集 $a_1a_2\cdots a_r$ 的后继者 $b_1b_2\cdots b_r$ 的方法是*从右到左*搜索直到找到第一个 $a_i\neq n-r+i$; 并对这个 i, 设 $b_i=a_i+1$, 而对于 $i<j<r$, 则设 $b_j=b_{j-1}+1$.

通常, 我们把最后的数字增加 1 就可以得到词典顺序下的下一个 3 元子集. 然而, 如果最后的数字是 6, 那么我们在查看前面的各数字之前, 就不能指定新的最大

数字. 如果中间的数字是 5, 那么必须转到第一个数字. 因为在打印最后的数字, 也可能在打印中间数字时有延迟, 所以我们输入的每一个数字后面插入两个空白, 记作–. 而且 (根据上面算法的需要) 从第 3 个数字 (最大的数字) 开始, 我们以相反顺序读入子集.

表 10.5 给出要求的有限状态机结构, 其中 t_0 是开始状态. 如果最后的数字 a_3 不是 6, 那么后继者 $b_3 = a_3 + 1$ 是最后的数字, 而且机器进入状态 t_1, 此时重复当前子集的其他两个数字, 它们不发生变化. 如果 a_3 是 6, 那么进入状态 d_1(延迟一轮), 并打印一个空格. 如果第二个数字 a_2 不是 5, 那么 $b_3 = a_2 + 2$, 且新的第二个数字是 $b_2 = a_2 + 1$. 如果 $a_2 = 5$, 那么进入状态 d_2(推迟两轮打印), 而此时新子集是 a_1+1, a_1+2, a_1+3.

表　10.5

	–	1	2	3	4	5	6
t_0	$t_0/-$			$t_1/4$	$t_1/5$	$t_1/6$	$d_1/-$
t_1	$t_1/-$	$t_1/1$	$t_1/2$	$t_1/3$	$t_1/4$		
d_1	$d_1/-$		$s_3/4$	$s_4/5$	$s_5/6$	$d_2/-$	
d_2	$d_2/-$	$s'_3/4$	$s'_4/5$	$s'_5/6$			
s_2	$t_1/2$						
s_3	$t_1/3$						
s_4	$t_1/4$						
s_5	$t_1/5$						
s'_3	$s_2/3$						
s'_4	$s_3/4$						
s'_5	$s_4/5$						

例如, 对于子集 246, 我们做如下处理 (输入是 6-4-2-):

$$(t_0, -) \xrightarrow{6} (d_1, -) \xrightarrow{-} (d_1, -) \xrightarrow{-} (d_1, -) \xrightarrow{4} (s_5, 6) \xrightarrow{-} (t_1, 5) \xrightarrow{-}$$

$$(t_1, -) \xrightarrow{2} (t_1, 2) \xrightarrow{-} (t_1, -) \xrightarrow{-} (t_1, -)$$ □

当 n 和 r 变大时, 这样设计的用于寻找 n 元集合下一个 r 元子集的机器就会变得非常复杂. 在内存中不能存储变量通常是有限状态机的巨大障碍. 然而, 有限状态机对于完成例 2 提出的任务类型 (识别正则语言中的句子) 还是相当优秀的.

为了更精确, 当下面的情况成立时有限状态机 M**识别**语言 L: 当 M(从一个指定的开始状态开始) 处理输入串 ω 时, 打印的最后输出元素是特定的 “识别” 元素, 当且仅当 ω 是 L 的句子.

定理　下面关于序列集合 L 的陈述等价.

(a) 对于某个正则文法 $G, L = L(G)$.

(b) L 被某个有限状态机识别.

我们只证明 (a) 的一种特殊情况蕴含 (b). 对于本书来说, 完整的证明太过复杂. 关于完整证明, 可以参见 Hopcroft, Motwani, and Ullman[1]. 正则文法 $G=(N,T,P,s)$ 是**简单的**(simple), 如果对于每一个 $t\in N$ 和 $a\in T$, 在 P 中至多存在一个形如 $t\to at'$ 的产生式 (不能存在 $a\to at'$ 且 $a\to at''$).

引理 如果 $L(G)$ 是简单正则文法 G 的语言, 那么存在识别 $L(G)$ 的有限状态机.

证明 设 G 的非终结符是 M 的状态, 而终结符是输入. G 的开始符号 s 是 M 的开始状态. 对于产生式规则 $t_i\to a_kt_j$, 定义 $f(a_k,t_i)=t_j$. 另外, 如果不存在产生式 $t_i\to a_k$, 则句子不可能完成, 所以定义 $\theta(a_k,t_i)=0$, 这是一个不 “识别” 输出元素. 如果存在产生式 $t_i\to a_k$, 那么一个句子可以 (但不一定) 在此结束, 于是定义 $\theta(a_k,t_i)=*$, 这是一个 “识别” 元素. 如果存在产生式 $t_i\to a_k$, 但是对某个 t_j, 不存在形如 $t_i\to a_kt_j$ 的产生式, 那么仍定义 $\theta(a_k,t_i)=*$, 但是此时 $f(a_k,t_i)=r$, 句子必须结束时 M 进入的一种特殊状态.

如果 M 进入状态 r 后仍有输入, 那么 M 打印 0 并永远停留在状态 r(打印 0). 如果在状态 t_i, M 读入 a_k, 但 G 没有 t_i 在左边而 a_k 在右边的产生式时, 我们也进入状态 r. 注意, 因为 G 是简单的, 所以 f 是定义明确的; 即, 至多存在一个给定的 t_i 在左边, 而 a_k 在右边的产生式. ◇

例 2 中的机器与 10.2 节中的文法 G_2 就有这种关系 (读者应该自行证明).

练 习 10.3

练习概况 前 3 道练习涉及跟踪带有给定输入的机器的动作. 练习 4 给出一个简单但重要的有限状态机的图模型. 其余练习都是有关构建执行各种功能的机器, 这些功能大部分是识别正则语言的.

1. 当例 1 中的机器处理下面输入时, 跟踪其后继内部状态和输出:
(a) 1100110; **(b)** 1010101.

2. 当例 2 中的机器处理下面输入时, 跟踪其后继内部状态和输出:
(a) 10101; **(b)** 1010011.

3. 当例 3 中的机器依词典顺序计算{1, 2, 3, 4, 5, 6}的下一个 3 元子集时, 跟踪其后继内部状态和输出:
(a) 135; **(b)** 136; **(c)** 356.

4. 有限状态机可以用有向多重图来表示, 其中, 每种内部状态都用一个顶点来表示. 如果 $f(s,a)=s'$ 且 $\theta(s,a)=o$, 那么存在一条有标签 a,o 的从 s 到 s' 的有向边. 画出与下面有限状态机有关的有向图.
(a) 例 1 中的机器. **(b)** 例 2 中的机器. **(c)** 例 3 中的机器.

5. 构建一个有限状态机, 它读入 $+1$ 和 -1 的序列, 并打印这个序列至此为止的和 (假设这个序列的和总是在 3 和 -3 之间, 包括 3 和 -3).

6. 构建一个有限状态机, 它计数三进制序列输入中子序列 012 的出现次数 (最多出现 3 次).

7. 构建一个有限状态机, 它读入数字 0 到 5 之间的数字序列 (不包含 0 和 5), 并打印至此所读入的数的和模 6.

8. 构建一个有限状态机, 它读入二进制表示的整数, 并打印二进制表示的下一个整数. 假设我们以相反顺序 (从最低位的数字开始) 读入这个二进制表示的整数.

9. 构建一个有限状态机, 它读入二进制表示的整数 n, 并打印二进制表示的整数 $n+3$. 假设我们以相反顺序读入这个二进制表示的整数.

10. 构建一个有限状态机, 它 (用例 3 的方式) 计算下列各集合的依词典顺序的下一个结果.

(a) {1, 2, 3, 4, 5}的 2 元子集.

(b) 1, 2, 3 的排列.

(c) {1, 2, 3}的重复 3 元子集.

11. 构建一个有限状态机, 它识别 10.2 节中给出的下面正则文法的语言.

(a) G_3'. **(b)** 练习 7(a).

10.4 小结及参考文献

本章介绍了 3 种借助计算机科学研究枚举的方法. 在现代数字计算机出现之后, 我们才有可能系统地处理枚举排列、选取以及其他组合集合的问题. 在 20 世纪 50 年代, Lehmer 是首批研究这种枚举问题的人之一, 见文献 [2]. 有关枚举置换和组合的算法的更多信息可以参见 Skiena[3].

我们利用形式语言和文法生成了各种组合集合的所有结果. 文法使第 5 章所使用的生成各种结果的计数过程形式化. 它们还给出计数问题中的递归关系. 在计算机科学中, 文法使程序设计语言的结构系统化.

有限状态机是计算机的最简单模型之一. 但是它们却很强大, 足以识别任意的正则语言. 正则语言与有限状态机之间的这种关系是理论计算机科学的诸多基础成果之一, 而且拥有广泛的实际效应. 对文法和有限状态机的更深入了解, 可以参见文献 [1].

1. J. Hopcroft, R. Motwani, and J. Ullman, *Introduction to Automata Theory, Languages, and Computation,* 2nd ed., Addison-Wesley, Reading, Ma., 2000.

2. D. Lehmer, "The Machine Tools of Combinatorics," in *Applied Combinatorial Mathematics,* E. Beckenbach(ed.), John Wiley & Sons, New York, 1964.

3. S. Skiena, *The Algorithm Design Manual*, Springer-Verlag, New Yok, 1997.

第11章　图　游　戏

11.1　逐步受限游戏

本章研究逐步受限游戏的理论, 并把它运用于尼姆类游戏中. 尽管这类游戏并非重要的图应用, 但是对这类游戏的图论分析产生了一些有趣的结果. 我们期望本章成为继前几章的若干较严肃课题的研究之后的一种数学“甜点”. 研究如何使用图模型来解决现实世界中的各种问题是十分重要的, 然而, 学习如何使用图以在某些游戏中取胜是对我们自己的奖赏.

两个玩家轮番做移动, 直到其中的一个玩家取胜为止 (不允许平局) 的游戏称为**逐步受限** (progressively finite) 游戏, 如果: (1) 在这个游戏中有有限个不同局势 (position); (2) 游戏必须在有限次移动后停止.

本节的目标是在游戏中取胜: 如何为逐步受限游戏确定取胜策略. 可以用下面的有向图来模拟逐步受限游戏: 顶点代表游戏进行中可能出现的局势, 而有向边代表从一种局势到另一种局势的可能移动.

观察发现, 逐步受限游戏图不能包含有向回路, 因为否则的话玩家就可能无休止地沿局势回路重复下去 (这违背有限次移动的限制). 因此, 在游戏的进行中, 任何局势都不能重复. 玩家必须严格地向着某个玩家取胜的最终局势移动. 诸如西洋跳棋和国际象棋这样允许平局及局势重复的游戏不是逐步受限游戏. 大多数逐步受限游戏是下面例 1 和例 2 说明的“取子”类游戏.

例 1　受限取子游戏

一张桌子上放置了 16 个对象. 两个玩家轮番移走 1、2、3 或 4 个对象. 移走最后一个对象的玩家是胜者. 这个游戏的游戏图如图 11.1 所示 (所有边都是从左到右的有向边). □

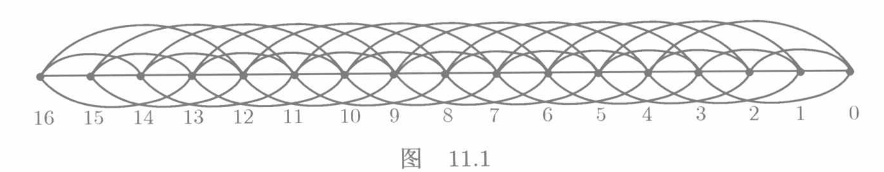


图　11.1

例 2　反取子游戏

从一个空堆开始, 两个玩家向这个空堆加入 1 枚一分硬币或 2 枚一分硬币, 或者 1 枚五分硬币, 直到这个堆的值是某个大于等于 2 的正整数的平方, 或者这个值超过 40. 加入硬币使这个堆的值达到临界量中的一个的玩家拿走这个堆中的所有

钱. 这个游戏的游戏图如图 11.2 所示 (所有边都是从左到右的有向边; 箭头指向取胜局势). 这个游戏是下面 “取子” 游戏的反向形式: 从一个堆中拿走 1、2 或 5 个对象, 直到这个堆的值减小到某个临界大小. □

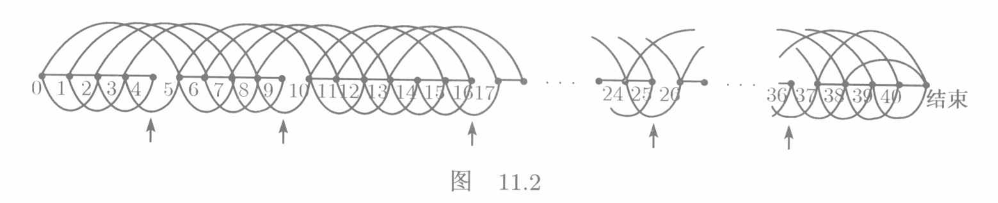


图　11.2

建议读者与朋友一起实际玩一下这样的游戏. 稍微考虑一下就会发现例 1 中的游戏的第一位玩家的取胜策略. 例 2 中的游戏稍困难一些. 它的细节只能使我们系统地寻找必胜策略的想法越发混乱. 开发逐步受限游戏的一般必胜策略理论, 然后再把这一理论运用到例 2, 这样会使问题较为容易一些.

逐步受限游戏的**取胜局势** (winning position) 是, 游戏在这样的局势下停止, 而且到达这个局势的玩家一定是胜者.

在逐步受限游戏图中, 出度为 0 的顶点一定表示取胜局势, 因为如果没有边离开一个顶点, 那么游戏一定在那个顶点停止. 我们称这样的顶点是**取胜顶点** (winning vertice). 例 1 中的游戏图只有一个标号为 0 的取胜顶点, 而例 2 的游戏图有 6 个取胜顶点, 标号分别为 4, 9, 16, 25, 36, 以及 “超过 40”. 每一个逐步受限游戏至少有一个取胜局势, 否则游戏将无休止地进行下去.

一个玩家的**取胜策略** (winning strategy) 是一种规则, 它告诉这个玩家在这个游戏的每一个阶段做哪一步移动, 从而确保这个玩家最终取胜, 即, 他最后进入一个取胜局势. 显然只有一个玩家有取胜策略.

任意与取胜顶点相邻的顶点在下面的意义下是一个失利顶点: 如果一个玩家移动到一个失利顶点, 那么另一个玩家就可以移动到一个取胜顶点 (并取得胜利). 在例 1 的游戏中, 标号为 1, 2, 3, 4 的顶点是失利顶点. 在例 2 的游戏中, 标号为 2, 3, 7, 8, 11, 14, 15, 20, 23, 24, 31, 34, 35, 37, 38, 39, 40 的顶点是失利顶点.

从例 2 的游戏结束点向回移动一步或更多步, 我们会看到: 如果所有边都从顶点 x 进入失利顶点, 那么 x 是一个 “预胜 (prewinning)” 顶点. 无论什么时候玩家 A 移动到这样的一个预胜顶点, 那么玩家 B 在下一次移动一定进入一个失利顶点, 而这时玩家 A 就能够进入一个取胜顶点. 例 1 中的游戏的顶点 5 就是一个预胜顶点, 因为所有边都从 5 进入失利顶点 (1, 2, 3, 4). 在例 2 的游戏中, 顶点 6 和 33 是预胜顶点, 而与失利顶点相邻的所有其他顶点同时也与非失利顶点相邻.

逐步受限游戏取胜策略的一般理论是建立在前面推理的递归扩展的基础上的. 我们要寻找诸如取胜顶点或预胜顶点这样的一些好顶点, 它们可以取胜或只导致到坏顶点的移动. 当对手被迫移动到一个坏顶点, 诸如失利顶点时, 那么我们将总能

够移动到另一些好顶点. 第一个玩家的取胜策略将告知这个玩家如何寻找从任意一个坏顶点能够到达的好顶点. 第一个玩家持续地进入好顶点, 而第二个玩家则被迫进入坏顶点的这种游戏模式一直进行下去, 直到第一个玩家进入一个是取胜顶点的好顶点为止. 如果一个游戏是从一个好顶点开始的, 那么第一个玩家的初始移动必须进入一个坏顶点, 这时整个游戏的角色就会反过来, 第二个玩家有使用好顶点的取胜策略.

例 1(续) 在这个游戏中, 第一个玩家的取胜策略是移动到一个标号为 5 的倍数的顶点上去. 这些都是好顶点. 因此, 从起始顶点 16 移动到 15(即第一个玩家拿走一个对象). 从 15 开始, 第二个玩家只能移动到 11, 12, 13, 14 中的一个顶点. 从这些坏顶点的任意一个顶点开始, 第一个玩家移动到顶点 10. 无论第二个玩家下一次移动到什么地方, 第一个玩家总能够移动到预胜顶点 5, 再进行一轮之后, 第一个玩家获胜. 注意, 如果这个游戏开始时只有 15 个对象, 那么游戏中的角色就会发生逆转, 第二个玩家将可以使用这一取胜策略. □

我们使用下面的定义把好顶点的概念形式化. 有向图的**核** (kernel) 是这样的顶点的集合:

(1) 没有连接核内任意两个顶点的边;

(2) 对于每一个非核顶点, 都存在一条从它到某个核顶点的边.

例 1 的核是标号为 0, 5, 10, 15 的顶点的集合.

定理 1 *如果逐步受限游戏图有核 K, 那么第一个玩家的取胜策略是每一轮都移动到一个核顶点. 然而, 如果起始顶点在核内, 那么第二个玩家可以使用这一取胜策略.*

证明 首先证明所有取胜顶点一定在 K 中. 其理由是对于不在核内的顶点, 一定有一条通向核内某个顶点的边. 但是取胜顶点的出度是 0. 因此, 所有取胜顶点都在核内.

接下来, 我们证明对第一个玩家来说, 每一轮到核内顶点的移动是取胜策略. 根据核的性质 (1), 当第一个玩家移动到核内的一个顶点时, 接下来第二个玩家必定移动到一个非核顶点. 于是根据性质 (2), 第一个玩家总能够从这个非核顶点移动到一个核内顶点. 这一游戏过程总是以第一个玩家移动到一个核内顶点, 而第二个玩家则总是移动到一个非核顶点的方式进行. 因为这一游戏是逐步受限的, 所以游戏最终一定会停止. 因为所有取胜顶点都在核内, 所以第一个玩家一定可以获胜.

如果起始顶点在核内, 那么第一个玩家和第二个玩家的角色要发生逆转, 第二个玩家总可以移动到核内的顶点, 因此最终取胜. ◇

前面的证明并没有明确地表明持续地到核内顶点的移动是如何到达一个取胜顶点的. 这一证明是存在的. 只需证明通过到核内顶点的移动, 第一个玩家最终一定可以到达一个取胜顶点.

一个更直接的问题是要证明每一个逐步受限游戏图都有一个核, 进而要找到这

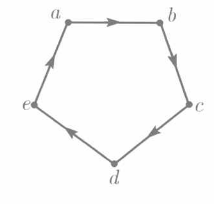


图 11.3

个核. 并不是所有图都有核. 例如, 图 11.3 中的图就没有核. 为了证明这个图没有核, 我们论证如下: 如果它有一个核, 那么可以通过这个图的对称性假设顶点 a 在核内. 有一条到达顶点 a 的顶点 e 就不能在核内. 顶点 d 唯一向外的边进入到顶点 e, 所以顶点 d 必定在这个核内. 类似地, c 不能在这个核内, 于是 b 一定在这个核内. 于是边 (a,b) 连接了核内的两个顶点, 矛盾.

万幸的是, 逐步受限游戏图都有核. 为了说明在这样的图中的核的存在性, 我们需要根据逐步受限游戏中的顶点到取胜顶点的 "距离" 来把它们组成不同的层次. 可以如下递归地定义有向图中顶点 x 的层次 $l(x)$, 以及位于 $\leqslant k$ 层上的顶点集合 L_k. 设 $s(x) = \{y \mid x$ 有通向 y 的边$\}$是 x 的**后继者**(successor) 的集合. 于是有:

$l(x) = 0$ 当且仅当 $s(x)$ 为空, 且 $L_0 = \{x|l(x) = 0\}$

$l(x) = 1$ 当且仅当 $x \notin L_0$ 且 $s(x) \subseteq L_0$, 且 $L_1 = L_0 \cup \{x|l(x) = 1\}$

一般地, 有

$l(x) = k$ 当且仅当 $x \notin L_{k-1}$ 且 $s(x) \subseteq L_{k-1}$, 且 $L_k = L_{k-1} \cup \{x|l(x) = k\}$

观察发现, $L_k - L_{k-1}$ 是位于层次 k 上的所有顶点的集合.

可以证明 $l(x)$ 是有向图中从顶点 x 开始的最长的路径长度 (参见练习 12). 因为逐步受限图的所有路径都有有限长度, 所以从顶点 x 开始的最长路径有有限长度. 另外, 从顶点 x 开始的最长路径必须结束于一个出度为 0 的顶点; 否则这条路径一定会延长. 于是, 根据递归定义, 从出度为 0 的顶点开始, 层次数的分配最终可以达到所有顶点.

根据层次的这种定义, 位于层次 0 的每一个顶点都是取胜顶点, 而且层次 1 的每一个顶点都是失利顶点. 层次 2 上的顶点如果与层次 0 上的顶点相邻, 那么它也是失利顶点; 只有当它的所有后继者都在层次 1 上时, 它是一个预胜顶点. 一般地, 层次 $k(k > 0)$ 上的每个顶点必定与层次 $k-1$ 上的一个顶点相邻, 它也可能与更低层次上的其他顶点相邻, 但不能与层次 k (或者更大层次) 上的任意其他顶点相邻. 现在, 我们就可以证明逐步受限游戏的基本定理.

定理 2 *每一个逐步受限游戏都有唯一的取胜策略. 即, 每一个逐步受限游戏都有唯一的核.*

证明 我们通过对层次, 或更精确地说对集合 L_k 进行归纳来给出证明. 设 K_k 是 L_k 中的核顶点集合. 首先考虑集合 L_0. L_0 是由出度为 0 的顶点组成的. 这些顶点都是取胜顶点. 正如在定理 1 中所提到的那样, 所有取胜顶点一定在这个核内. 因此 $K_0 = L_0$.

接下来, 归纳地假设对于 $n \geqslant 1$, K_{n-1} 是唯一、定义明确的 L_{n-1} 中的核顶点集合. 我们证明能够找到层次 n 的顶点集合, 把这个集合加入到 K_{n-1} 中形成 L_n 的核 K_n. 根据层次的定义方式, $l(x) = n$ 表明 $s(x)$ 包含在 L_{n-1} 中. 如果一个层次 n 的顶点 x 不与 K_{n-1} 的核顶点相邻, 那么 x 一定在 K_n 中 (因为根据核的定

义, 其后继者不在核内的任意顶点一定在核内). 另一方面, 如果 x 与 K_{n-1} 的核内的一个顶点相邻, 那么 x 不可能在核内. 因此, $K_n = K_{n-1} \cup \{x \mid l(x) = n$ 且 $s(x) \cap K_{n-1} = \phi\}$ 是 L_n 的唯一、定义明确的核顶点集合. 于是, 根据数学归纳法, 这个图有唯一核. 根据定理 1, 这个核是唯一取胜策略. ◇

定理 2 的证明告诉我们如何构建核. 首先把取胜顶点放入核内. 然后, 递归地把这些顶点加入到不与核内顶点当前集合相邻的递增层次上. 我们利用称为 Grundy 函数的标签规则完成这一寻找核的过程.

Grundy 函数 $g(x)$ 的定义

对于有向图中的每一个顶点 x, $g(x)$ 是没有指定给 x 的任意后继者的最小非负整数.

我们将简略地证明: Grundy 数为 0 的顶点正好组成核顶点集合. 在 11.2 节, 我们将看到在更复杂的游戏中 Grundy 数充当着基础的角色.

在逐步受限游戏图中, 利用层到层的方法很容易确定 Grundy 值. 层次 0 上的顶点没有后继者, 所以它们的 Grundy 数是 0(最小非负整数). 下面, 我们确定层次 1 上的顶点 x 的 $g(x)$, 然后层次 2 上的顶点, 等等. 这样, 在确定 x 的 Grundy 数之前, 它的所有后继者都分配了 Grundy 数, 因为它的后继者位于较低的层次上. 当我们进行到 x 时, 我们可以检查 x 的后继者 $s(x)$ 的 Grundy 函数值, 并设 $g(x)$ 等于没有分配给 $s(x)$ 中任意顶点的最小非负整数. 实际上, 我们不需要完全一层一层地进行. 可以使用直到一个顶点的所有后继者都有 Grundy 数时才给它分配 Grundy 数的任意方法.

我们无法为图 11.3 定义 Grundy 函数. 图 11.3 中每一个顶点 x 都有一个后继者, 而这个后继者的 Grundy 数一定在 $g(x)$ 确定之前定义. 即使我们尝试着一次给图 11.3 的所有顶点创建 Grundy 数, 也不存在 Grundy 函数 (其细节留给读者).

下表给出了图 11.1 和图 11.2 的顶点的 Grundy 数. 注意, 图 11.1 中的核内顶点是 0, 5, 10, 15, 它们的 Grundy 数是 0. 这并非偶然.

图 11.1 的 Grundy 数

顶点 x	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
$g(x)$	1	0	4	3	2	1	0	4	3	2	1	0	4	3	2	1	0

定理 3 *逐步受限游戏图有唯一的 Grundy 函数. 而且, Grundy 数为 0 的顶点是核中的顶点.*

证明 如上所述, 可以一层一层递归地构建逐步受限游戏图的 Grundy 函数, 从而给每一个顶点唯一一个 Grundy 数. 根据 Grundy 函数的定义, Grundy 数为 0 的顶点 x 不能有 Grundy 数为 0 的后继者 (否则 x 必须有一个不同的数). 类似地, Grundy 数为 $g(y) = k > 0$ 的任意顶点 y 必定有一条边通向 Grundy 数为 0 的某个

顶点 (否则 y 的数等于 0). 因此, Grundy 数为 0 的顶点的集合满足核的两个定义性质. ◇

图 11.2 的 Grundy 数

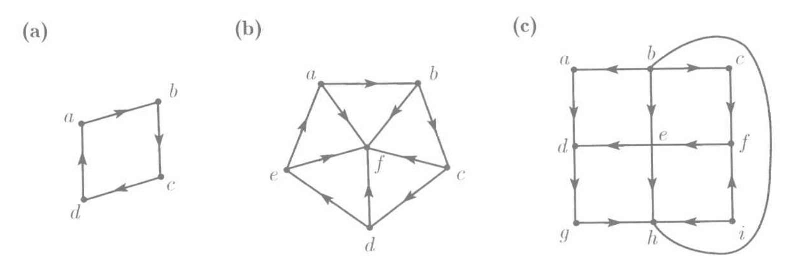
顶点 x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
$g(x)$	0	3	1	2	0	1	0	2	1	0	3	1	0	3	1	2	0
	17	18	19	20	21	22	23	24	25	26	27	28	29	30			
	1	0	2	1	0	3	2	1	0	2	0	1	2	0			
	31	32	33	34	35	36	37	38	39	40	大于 40						
	1	2	0	1	2	0	1	3	2	1	0						

例 2(续)　根据定理 3 和图 11.2 的 Grundy 数的表格, 我们看到这个游戏的核是集合 0, 4, 6, 9, 12, 16, 18, 21, 25, 27, 30, 33, 36, "超过 40". 因为起始顶点在核内, 所以第二个玩家在这个游戏中有取胜策略. 这个游戏可以如下进行 (设 A 是第一个玩家, B 是第二个玩家): 首先 A 移动到 1, 然后 B 移动到核顶点 6, 那么 A 必须到顶点 11(因为他移动到 7 或者 8, B 在 9 都取胜), 然后 B 移动到核顶点 12, 而 A 必须移动到 17, 然后, B 移动到核顶点 18, 然后 A 移动到 20, 然后 B 移动到取胜顶点 25(并拿走全部 25 美分). □

练　习　11.1

练习概况　前 10 道练习主要是求各种逐步受限游戏图的核及 Grundy 函数. 练习 11~17 是核、Grundy 函数及层次数的性质的证明. 练习 18 和 19 给出两个更复杂的逐步受限游戏.

1. 求下列各图的核, 或者说明为什么不存在核:



2. 假设有 25 根小棍, 两个玩家轮番一次移走最多 5 根小棍. 赢家是移走最后一根小棍的玩家. 哪个玩家有取胜策略? 描述这一策略.

3. 重复例 2, 此时玩家每次加入 2、3 或 7 美分. 求核中位置的集合.

4. 证明: 在例 2 中, 第二个玩家总可以在第二次移动取胜.

5. 假设在例 2 中, 第一个玩家 A 知道他/她将失利, 因此想要使其损失最小. A 逼迫 B(第二个玩家) 接受 (使得如果 B 想使游戏继续进行下去, 那么 A 将进入到核内并取胜) 的最小取胜量是多少?

6. **(a)** 假设我们有一堆 7 根红棍和 10 根蓝棍. 一个玩家可以移动任意数量的红棍, 或者任意数量的蓝棍, 或者移动等量的红棍和蓝棍. 赢家是移动最后一根棍的玩家. 求核中位置的集合.

(b) 如果每一次移动限制为一种颜色 (或两种颜色) 的棍数不能超过 5, 重复部分 (a).

7. 重复例 1, 但是此时, 移走最后一个对象的玩家是失败者. 描述这个游戏的取胜策略.

8. 求下面的图或游戏的 Grundy 函数:

(a) 练习 1(a); **(b)** 练习 1(b); **(c)** 练习 3.

9. 证明图 11.3 不存在 Grundy 函数.

10. 求一个有向图, 它拥有核但是没有 Grundy 函数.

11. 如果 $W(S)$ 是没有连接到 S 中任意顶点的边的顶点集合. 证明: 集合 S 是一个核当且仅当 $S = W(S)$.

12. **(a)** 证明: 对于逐步受限图 G 中的顶点 x, 如果 $l(x) = k$, 那么 k 是 G 中开始于 x 的最长路径的长度.

(b) 证明: 如果 $g(x) = k$, 那么在 G 中存在一个开始于 x 的长度为 k 的路径.

13. 证明: 对于逐步受限图中的任意顶点 x, 有 $g(x) \leqslant l(x)$.

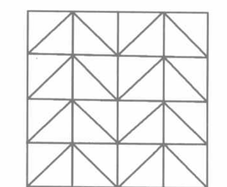
14. 证明: 如果有向图 G 和 G 的每一个 (删除其中某些顶点而得到的) 子图都有核, 则 G 有 Grundy 函数.

15. 证明: 逐步受限图 G 的层次数和 Grundy 函数构成 G 的正确着色.

16. 证明: 无论怎样为二部图中的边定向, 该二部图总是有 Grundy 函数.

17. 证明: 逐步受限游戏图只能有有限个顶点. [提示: 证明如果顶点数量是无限的, 那么一定存在一条无限长的路径 (游戏可以无限进行下去).]

18. 考虑下面的图游戏. 玩家 A 使用右图中从左到右的一组顶点, 尝试着做一条从这个图的左边到右边的路径. 玩家 B 尝试着做一条从上到下的路径. 这两个玩家轮番选择顶点直到一个玩家得到一条想要的路径.



(a) 证明: 这个游戏是逐步受限游戏.

(b) 求玩家 A 的取胜策略.

19. kayles 游戏有一行 n 个等间隔石子. 两个玩家交替移走一个或两个 (中间没有空间的) 连续石子. 移走最后一个石子的玩家是胜者.

(a) 画出 4 个石子的游戏图, 并求它的 Grundy 函数.

(b) 对于 6 石子游戏, 重做问题 (a).

11.2 尼姆类游戏

本节把逐步受限游戏的理论推广到包含若干堆对象的取子游戏. 这类游戏中最简单的是尼姆游戏: 两个玩家轮番从几堆对象之一移走任意数量的对象. 胜者是从最后剩余的一堆 (非空的) 对象中移走最后一个对象的玩家. 尽管 11.1 节例 1 和例 2 中的两个游戏的局势可以用代表单一堆的大小或钱数的非负整数表示, 而在尼姆游戏中, 局势要求非负整数的向量 $(p_1, p_2, \cdots, p_m)$, 其中第 k 个数表示第 k 堆的当

前大小.

例 1 尼姆游戏

图 11.4

考虑有 4 堆小棍的尼姆游戏: 第一堆中有 1 根小棍, 第二堆里有 2 根小棍, 第三堆里有 3 根小棍, 第四堆里有 4 根小棍. 参见图 11.4. 我们用向量 (1, 2, 3, 4) 表示这个游戏的初始局势.

设第一个和第二个玩家分别命名为 A 和 B. A 可以如下这样玩这个游戏. 首先, A 移走第四个堆里的所有 4 根小棍. 新局势是 (1, 2, 3, 0). 接下来, B 移走第三堆中的一根小棍, 结果产生局势 (1, 2, 2, 0). 现在, A 移走第一堆里的一根小棍, 产生局势 (0, 2, 2, 0). A 是在执行一个取胜策略, 即移进到核位置, 而且现在就要赢了. 如果 B 移走第二堆或者第三堆的所有小棍, 那么 A 将移走其他堆, 或者如果 B 只移走一根小棍, 那么 A 将从另一个堆移走一根小棍, 于是 A 将在下一轮取胜. 我们希望读者与朋友一起玩一玩这种游戏. □

例 1 中的游戏有 2×3×4×5=120 不同局势 (第 i 堆有 $i+1$ 个可能大小: $0,1,\cdots,i$). 因此, 画出这个游戏的图, 并像 11.1 节那样计数 Grundy 函数和它的核是一件非常麻烦的事. 一般地, 有 m 堆且第 i 堆有 n_i 个对象的尼姆游戏有 $(n_1+1)(n_2+1)\cdots(n_m+1)$ 个不同的局势. 不使用计算机, 在尼姆游戏中取胜的唯一方法是直接通过位置向量 $(p_1,p_2\cdots,p_m)$ 来确定局势 (向量) 的 Grundy 数. 幸运的是, 这样的直接计算是可能的.

首先, 我们需要看一下单堆尼姆游戏的 Grundy 函数. 考虑有一堆 4 根小棍的尼姆游戏. 参见图 11.5. 对于所有的 $0\leqslant j\leqslant i\leqslant 4$, 这个游戏图有边 $(i\to,j)$. 取胜顶点 0 的 Grundy 数是 0, 顶点 1 的 Grundy 数是 1, 等等. 在任意单堆尼姆游戏中, 顶点 i 的 Grundy 数是 i, 因为顶点 i 的边都指向所有标号更低的顶点. 因此, 任意单堆尼姆游戏的策略是显然的: 第一个玩家移走整个堆, 而且获胜. 这种单堆 Grundy 函数的好形式可以简化多堆尼姆游戏的 Grundy 函数的计算.

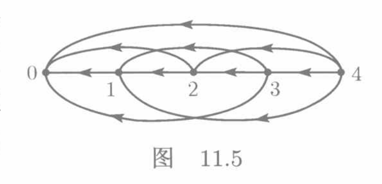


图 11.5

尽管多堆尼姆游戏图复杂且不易画出, 但是我们仍可以象征性地描述它. 对每一种局势 $(p_1,p_2,\cdots,p_m)$, 可以使用一个顶点来表示它, 其中, $0\leqslant p_i\leqslant n_i$ (n_i 是第 i 堆的初始大小). 因为允许的移动是从一堆中移走一定量的对象, 因此, 对每一对顶点 $(p_1, p_2, \cdots, p_m)$ 和 $(q_1,q_2,\cdots,q_m)$, 其中对某个 j, 有 $q_j<p_j$, 且对所有的 $i\neq j, q_i=p_i$, 相关图有一条从顶点 $(p_1,p_2,\cdots,p_m)$ 到 $(q_1,q_2,\cdots,q_m)$ 的边. 在某种意义上, 这个图是每一堆的图的合成, 因为除了一个 p_i 之外, 比如说是 p_3, 我们固定所有其他的 p_i, 那么顶点 $(p_1,p_2,0,p_4\cdots,p_m)$, $(p_1,p_2,1,p_4,\cdots,p_m),\cdots,(p_1,p_2,n_3,p_4,\cdots,p_m)$ 所形成的子图正好是堆 3 的图.

这种类型的图合成可以如下形式化. 分别拥有顶点集合 $X_1, X_2,\cdots,X_m$ 的图

$H_1, H_2, \cdots, H_m$ 的**直和** (direct sum) $H = H_1 + H_2 + \cdots + H_m$ 有顶点集合 $X = \{(x_1, x_2, \cdots, x_m) \mid x_i \in X_i, 1 \leqslant i \leqslant m\}$, 而它的边是由下面的后继者集合来定义的:

$$\begin{aligned} s((x_1, x_2, \cdots, x_m)) = & \{(y, x_2, x_3, \cdots, x_m)|y \in s(x_1)\} \\ & \cup\{(x_1, y, x_3, \cdots, x_{m1}|y \in x(x_2)\} \\ & \vdots \\ & \cup\{(x_1, x_2, \cdots x_{m-1}, y)|y \in s(x_m)\} \end{aligned}$$

根据这一定义, m 堆尼姆游戏的图 G 是第 i 堆的图 G_i 的直和:

$$G = G_1 + G_2 + \cdots + G_m$$

例 2　2×2 尼姆图

考虑有两堆且每一堆有两个对象的尼姆游戏. 图 11.6a 给出单堆的图 $G_1 = G_2$. 图 11.6b 给出两堆游戏图 $G = G_1 + G_2$. 求这个图的 Grundy 函数和核.

图 11.6b 给出各顶点的 Grundy 数. Grundy 数等于 0 的顶点在核内. 因为 $g((2,2)) = 0$, 所以根据 11.1 节的定理 1 和定理 3, 我们知道第二个玩家有取胜策略.

□

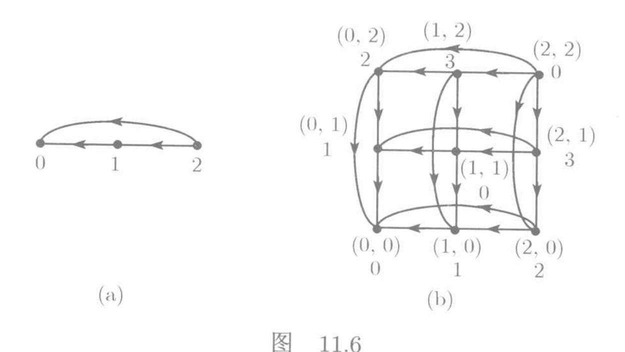


图 11.6

接下来说明如何根据各部分图 G_i 中的各顶点 x_i 的 Grundy 数 $g(x_i)$, 计算这些图的直和中的顶点 $(x_1, x_2 \cdots, x_m)$ 的 Grundy 数. 在这些 Grundy 数上所进行的计算称为数字和.

非负整数 $c_1, c_2, \cdots, c_m$ 的**数字和** (digital sum) c, 记作

$$c = c_1 \dotplus c_2 \dotplus c_3 \dotplus \cdots \dotplus c_m$$

是以下面的方式进行计算的. 设 $c^{(k)}$ 是 c 的二进制展开式中的第 k 个二进制数字, 即 $c = c^{(0)} + c^{(1)}2 + c^{(2)}2^2 + \cdots$, 类似地, 设 $c_i^{(k)}$ 是 c_i 的二进制展开式中的第 k 个二进制数字. 那么, $c_k = c_1^{(k)} + c_2^{(k)} + \cdots + c_m^{(k)}$(模 2). 即 c 的第 k 个二进制数是 1, 如果 c_j 的第 k 个二进制数字的和是奇数; c 的第 k 个二进制数字是 0, 如果 c_j 的第 k 个二进制数字的和是偶数.

	2^3	2^2	2^1	2^0
2 =	0	0	1	0
12 =	1	1	0	0
15 =	1	1	1	1
8 =	1	0	0	0
	1	0	0	1 = 9

图 11.7

例 3 数字和

计算数字和 $2\dot{+}12\dot{+}15\dot{+}8$. 我们把数字 2, 12, 15, 8 写成二进制的形式, 确定每一列上的数字和 (模 2), 如图 11.7 所示. 把二进制的和再转化成一个整数, 我们看到这个数字和等于 9. □

现在给出一个重要的定理. 本质上, 这个定理说数字和是在尼姆游戏中取胜的唯一方法.

定理 *如果图 $G_1, G_2, \cdots, G_m$ 有 Grundy 函数 $g(x)$, 那么, 直和 $G = G_1 + G_2 + \cdots + G_m$ 有下式定义的 Grundy 函数 $g(x)$, 其中 $x = (x_1, x_2, \cdots, x_m)$:*

$$g(x) = g((x_1, x_2, \cdots, x_m)) = g_1(x_1)\dot{+}g_2(x_2)\dot{+}\cdots\dot{+}g_m(x_m)$$

证明 我们必须证明 $g(x)$ 是不等于集合 $\{g(y) \mid y \in s(x)\}$ 中任意数的最小非负整数. 这一 Grundy 函数的定义可以分成两部分: (1) 如果 $y \in s(x)$, 那么 $g(x) \neq g(y)$; (2) 对于每一个非负整数 $b < g(x)$, 一定存在某个 $y \in s(x)$, 使得 $g(y) = b$.

部分 (1): 证明如果 $y \in s(x)$, 那么 $g(x) \neq g(y)$. 如果 $y = (y_1, y_2, \cdots, y_m) \in s((x_1, x_2, \cdots, x_m))$, 那么根据直和的定义, 对于某个 $j, y_j \in s(x_j)$, 且对于所有 $i \neq j, y_i = x_i$. 如果 $c_k = g_k(x_k)$ 且 $d_k = g_k(y_k)$, 那么 $c_j \neq d_j$ 且 $c_i = d_i$. 所以有:

$$g(x) = c_1\dot{+}c_2\dot{+}\cdots\dot{+}c_j\dot{+}\cdots\dot{+}c_m = c'\dot{+}c_j$$
$$g(y) = d_1\dot{+}d_2\dot{+}\cdots\dot{+}d_j\dot{+}\cdots\dot{+}d_m = c'\dot{+}d_j$$

其中, c' 是除 c_j 外的所有 c 的数字和. 不难证明 $c'\dot{+}c_j = c'\dot{+}d_j$ 当且仅当 $c_j = d_j$ (参见练习 9). 因为 $c_j \neq d_j$, 我们得出结论 $g(x) \neq g(y)$.

部分 (2): 证明对于每个非负整数 $b < g(x)$, 存在某个 $y \in s(x)$, 使得 $g(y) = b$. 这部分的一般证明非常有技巧 (其详细证明参见 Berge[2], p.25). 下面讨论的这个证明的实际运用是寻找使得 $g(y) = 0$ 的 (核顶点) y 并移动到它. 部分 (2) 的形式证明是下面讨论的一般形式. ◇

推论 *m 堆尼姆游戏图中的顶点 $(p_1, p_2, \cdots, p_m)$ 有 Grundy 数 $p_1\dot{+}p_2\dot{+}\cdots\dot{+}p_m$. 因此 $(p_1, p_2, \cdots, p_m)$ 是核顶点, 当且仅当 $p_1\dot{+}p_2\dot{+}\cdots\dot{+}p_m = 0$.*

读者应该返回到例 2 中每一堆有两个对象的两堆尼姆游戏, 并验证由那个游戏图的顶点而得到的 Grundy 数正好是堆大小的数字和.

例 1(续) 如图 11.8 所计算的那样, 图 11.4 中的尼姆游戏的起始顶点的 Grundy 数是数字和 $1\dot{+}2\dot{+}3\dot{+}4 = 4$. 第一个玩家 A 想要减少其中一堆的大小使得新的数字和等于 0(核位置). 即, A 应该改变图 11.8 的某行上的二进制数字, 使得和 (在底部)

的行是 0 0 0. 为了使和的行都变成 0, 此时, 应该改变有奇数个 1 的每一列的某一位数字 (或者是 1 变成 0 或者是 0 变成 1), 来使得那一列上的 1 的个数是偶数.

1	=	0	0	1
2	=	0	1	0
3	=	0	1	1
4	=	1	0	0
		1	0	0 = 4

图 11.8

因为正好在 2^2 列上和行有 1, 所以可以把那一列上的这一单个 1 变成 0. 即堆 4 的二进制展开应该从 100 变成 000, 所以玩家 A 应该移走第四堆上的所有小棍. 这个新局势 (1, 2, 3, 0) 的 Grundy 数是 0. 注意, 为了使 2^2 列上的 1 的个数成为偶数, 我们不能把任意的 0 变成 1, 因为改变后的行的新二进制展开将会是一个更大的数, 在尼姆游戏中这是一次不可行的移动. □

现在, 我们扩展前面例子中求 Grundy 数为 0 的顶点的方法. 给当前的游戏局势做一个如图 11.8 那样的数字和表. **挑出行 e, 它在最左边有奇数个 1 的列上有一个 1, 即, 行 e 在最左边的和行为 1 的列上有一个 1. 在行 e 上, 改变和行为 1 的每个列上的数字.** 经过这样的数字改变之后, 每个列都有偶数个 1, 所以和行都是 0. 因此, 新的局势是核顶点. 注意, 因为在行 e 中改变的最左边数字是 1(这就是为什么要选择 e), 所以行 e 上的数字改变产生一个更小的数 h. 因此, 第一个玩家应该把第 e 堆的大小减少到有 h 个对象的大小.

(a)	(b)	2^2	2^1	2^0
II	2 =	0	1	0
III	3 =	0	1	1
IIII	4 =	1	0	0
IIIIII	6 =	1	1	0
		0	1	1 = 3

图 11.9

例 4 另一个尼姆游戏

考虑图 11.9a 所示的分别有 2, 3, 4, 6 根小棍的四堆尼姆游戏. 图 11.9b 给出初始局势的数字和表. 2^1 列是和行有 1 的最左边的列, 所以必须改变在 2^1 列上有 1 的行. 可以使用第一行、第二行或第四行. 假设我们选择第一行. 于是, 因为 2^1 和 2^0 列在和行上有 1, 所以改变第一行上的这些列上的数字. 新的第一行是 001. 因此, 第一个玩家应该把第一堆的大小减少到 1. 读者应该与朋友玩一玩这一尼姆游戏来实际验证一下这种求核规则. □

练 习 11.2

1. 求下列各尼姆游戏初始局势的 Grundy 数, 并做取胜策略下的第一次移动.



2. 假设每次从任何一堆都不能移走多于两根小棍. 在这一附加条件下, 重复练习 1 中的各游戏.

3. 假设每次从第 i 堆中移走的小棍数目不能超过 i (从上到下给堆标号). 在这一附加条件下, 重复练习 1 的各游戏.

4. 假设从每一堆只能移走 1, 2, 5 根小棍. 在这一限制下, 重复练习 1 的各游戏.

5. 假设从每一堆只能移走 1 或 4 根小棍. 在这一限制下, 重复练习 1 的各游戏.

6. 对于练习 1(c) 的尼姆游戏, 求可以产生有下面的 Grundy 数的局势的移动:

(a) 1 **(b)** 2 **(c)** 3

7. 假设同时进行 3 个 11.1 节例 1 的游戏. 当一个堆的值是一个数的平方或超过 40 时, 玩家就停止向这个堆加钱币. 把最后的钱数加到最后一堆的玩家取胜, 拿走所有三堆的钱. 注意, 这个游戏有 11.1 节末尾给出的 Grundy 数表.

(a) 如果起初有两堆各有 2 美分, 而第三堆有 1 美分, 那么这种局势的 Grundy 数是什么? 正确的取胜移动是什么?

(b) 求前两堆的钱的总数少于 10 美分, 而第三堆为空的所有核位置.

(c) 使用 11.1 节末尾的单堆游戏的 Grundy 数表, 写出一个计算机程序来计算在这样的三堆游戏中的下一次取胜移动.

8. 画出下图的直和:

(a) G_1+G_1 **(b)** $G_1+G_2+G_3$

x_1 x_2 G_1 x_3 x_4 y_1 G_2 y_2 z_1 G_3 z_2

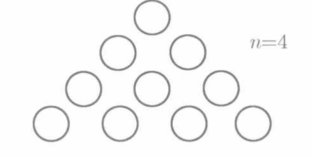
9. 证明: 在定理的证明中, $c'\dot{+}c_j = c'\dot{+}d_j$ 当且仅当 $c_j = d_j$.

10. 扩展例 4 中求尼姆游戏的核顶点的讨论, 得到定理证明的部分 (2).

11. 对于尼姆游戏, 考虑最后移动的玩家输的变形版本. 证明它的取胜策略是进行正规的最后移动玩家取胜的尼姆游戏策略, 但当只有一堆的小棍数目超过 1 时, 把这个堆的大小减少到 1 而不是减少到 0(或减少到 0 而不是 1).

12. 写出在尼姆游戏中取胜的计算机程序.

13. 考虑下面的变形版本的尼姆游戏. 把 $C(n+1,2)$ 个相同的球放置成一个三角形, 就像保龄球瓶那样. 两个玩家轮番移走位于一条直线上的一组球中的任意个球. 移走最后一个球的玩家取胜.



(a) 当 $n=3$ 时, 求在这个游戏中取胜的第一次移动.

(b) 当 $n=4$ 时, 重复部分 (a).

11.3 小结及参考文献

C. Bouton 于 1902 年出版了第一本尼姆分析的著作 (见文献 [3]). 逐步受限游戏的 Grundy 函数是由 P. Grundy 于 1939 年提出的 (见文献 [5]), 而德国数学家 Sprague 早在几年前就已经独立发现了这个函数.

本章只对逐步受限游戏理论做了一些粗浅的讨论. 有兴趣的读者可以查阅 Berlekamp, Conway, and Guy[1] 所著的《取胜方法》一书. 读者还可以参阅由 Coxeter 和 Ball 所著的关于数学和游戏的经典图书《数学娱乐和随笔》(见文献 [4]).

1. E. Berlekamp, J. Conway, and R. Guy, *On Winning Ways*, 2 volumes, Academic Press, New York, 1982.

2. C. Berge, *The Theory of Graphs*, Methuen-Wiley, New York, 1962.

3. C. L. Bouton, "Nim, a game with a complete mathematical theory," *Ann. Math.* 3(1902), 35–39.

4. H. Coxeter and W. Ball, *Mathematical Recreations and Essays*, University of Toronto Press, 1972.

5. P. M. Grundy, "Mathematics and games, " *Eureka*, January, 1939.

附 录 A

A.1 集 合 论

集合是一组不同对象. 与对象序列不同, 集合是无序的. 通常称集合中的对象为**元素**(element), 或**成员**(member). 这些元素本身也可以是集合, 如在所有 5 张手牌的集合中, 每一手牌都是一个 5 张牌的集合. 一组可重复出现的对象称为**族**(family), 例如, $\{a, a, a, b, b, c\}$.

集合 S 是集合 T 的子集, 如果 S 的每一个元素也是 T 的元素. 对于一手 5 张牌的情况, 我们使用所有 52 张牌的集合中的 5 张牌子集. 任意集合都是它自身的平凡子集. **真子集**(proper subset) 是元素比整个集合少的非空子集. 没有公共元素的两个集合称为**不相交**(disjoint) 集合.

本文使用大写字母表示集合, 小写字母表示元素 (除非元素本身是集合). 集合 S 中的元素个数表示为 $N(S)$ 或者 $|S|$. 符号 $\in$ 代表集合成员关系, 例如 $x \in S$ 表示 x 是 S 的元素, $x \notin S$ 表示 x 不是 S 的元素. 符号 "$\subseteq$" 代表子集包含关系, 例如 $T \subseteq S$ 表示 T 是 S 的子集.

用形式数字记法来定义集合有 3 种方法:

(1) 列出集合的元素, 如 $S = \{x_1, x_2, x_3, x_4\}$, 或 $T = \{\{1,-1\},\{2,-2\},\{3,-3\},\{4,-4\},\{5,-5\},\cdots\}$, 这表示 T 是正数和它的相反数的数对的集合.

(2) 通过定义性质, 如 $P = \{p \mid p$ 是选修这门课的人$\}$, 或 $R = \{r \mid$ 存在满足 $t \neq 0$ 且 $r = s/t$ 的整数 s 和 $t\}$, R 是所有有理数的集合.

(3) 其他集合上的操作结果 (见下面).

我们经常使用两个特殊的集合: **空集**(empty) 或**零集**(null), 写作 ϕ; 所考虑的所有对象的集合, **全集**(universal), 写作 $\mathscr{U}$.

我们应该记住, 在很多实际问题中, 我们无法精确地定义或列举集合. 例如, 大型计算机程序的可能失败方法这一集合是无法定义的. 美国对几个重要的平民亚类的人口普查存在实质性的错误, 而 1990 年官方的美国人口数是 248 709 873. 要对计算都是在不精确的集合上进行的加以警惕!

集合上的 3 个基本操作是:

(1) 集合 S 和 T 的**交集**(intersection), $S \cap T = \{x \in \mathscr{U} \mid x \in S$ 且 $x \in T\}$;

(2) 集合 S 和 T 的**并集**(union), $S \cup T = \{x \in \mathscr{U} \mid x \in S$ 或 $x \in T\}$;

(3) 集合 S 的**补集**(complement), $\overline{S} = \{x \in \mathscr{U} \mid x \notin S\}$.

第 4 个操作有时候很有用:

(4) 集合 S 减去集合 T 的**差**(difference), $S - T = \{x \in \mathscr{U} \mid x \in S$ 且 $x \notin T\}$;

观察发现, $S - T$ 可以用前面的操作表示成 $S - T = S \cap \overline{T}$.

例 1 选择计算器

	有内存	无内存
可充电	2	2
不可充电	2	2

图 A1.1

8 个不同品牌的计算器样本中, 4 台有充电电池, 另 4 台没有充电电池. 而且, 8 台计算器中有 4 台有内存. 我们想要选出 4 台计算器进行全面的分析. 所选出的 4 台机器中的一半应该有充电电池, 而一半应该有内存. 有多少种方法从这 8 个品牌的机器中选出 4 台机器?

为了回答这个问题, 我们必须知道有多少台机器可充电且有内存, 有多少台机器可充电但没有内存, 等等. 即, 如果 R 是可充电机器的集合, M 是有内存机器的集合, 那么我们需要知道 $N(R \cap M), N(R \cap \overline{M}), N(\overline{R} \cap M)$ 和 $N(\overline{R} \cap \overline{M})$ 等等. 假设这 4 个子

集中每一个都有两台机器. 参见图 A1.1. 那么选出 4 台机器的理想组合的策略之一是从图 A1.1 中的每一个范畴中选出一台机器 ($2\times2\times2\times2=16$ 种选择). 从图 A1.1 中选出 4 台机器还有其他两种办法 (参见本节末尾的练习 4). □

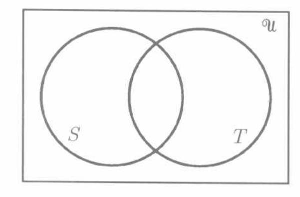
涉及前述集合操作以及相关规则的集合表达式的研究称为**布尔代数**(Boolean algebra). 布尔代数的 3 个最重要的规则是

BA1. $S=\overline{\overline{S}}$;

BA2. $\overline{S\cap T}=\overline{S}\cup\overline{T}$;

BA3. $\overline{S\cup T}=\overline{S}\cap\overline{T}$.

为了可视化集合表示, 我们使用 Venn 图 (Venn diagram). 图 A1.2 给出集 S 和 T 的 Venn 图. 整个矩形表示所考虑的所有元素的全集 $\mathscr{U}$. 圆表示集合 S 和集合 T. 在图 A1.2b 中, 涂黑的区域代表 $S\cap T$, 而有条纹的区域是 $(S\cup T)-(S\cap T)$. 注意在特殊的问题中, Venn 图中的某些区域可能是空集.



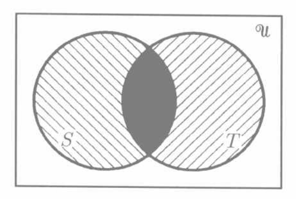


图 A1.2

下面的例子说明规则 BA2 和 BA3 是如何简化计数问题的.

例 2 利用布尔代数计数

考虑从一副 52 张牌中选出一张牌, 然后再放回这张牌, 再从中选出 (可能是相同的) 第二张牌的所有 52^2 个结果的全集 $\mathscr{U}$. 假设我们想要计算至少有一张黑桃或一张红心的所有结果集 Q 的大小. 设 S 是没有黑桃的所有结果集合, 即两次挑选都不选取黑桃. 设 H 是没有红心的结果的集合. 写出用 S 和 H 表示的集合 Q, 并计算 $N(Q)$.

集合 Q 等于 $\overline{S}\cup\overline{H}$, 这是有一张或多张黑桃的结果集合 (有黑桃) 与有一张或多张红心的结果集合的并集. 集合 $\overline{S}\cup\overline{H}$ 是图 A1.3 中的阴影区域. 根据 BA3, $\overline{S}\cup\overline{H}=\overline{S\cap H}$, 其中 $S\cap H$ 是没有黑桃且没有红心的结果集合 ($S\cap H$ 是图 A1.3 中的无阴影区域). 即 $S\cap H$ 是每次从 26 张梅花和方片中选出一张牌的结果集合. 因此 $N(S\cap H)=26^2$. 根据图 A1.2, $N(\overline{S\cap H})=N(\mathscr{U})-N(S\cap H)=52^2-26^2=2\,704-676=2\,028$, 所以 $N(\overline{S}\cup\overline{H})=2\,028$. □

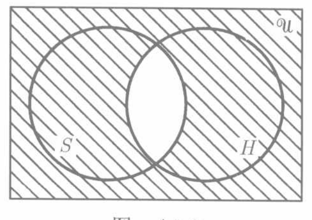


图 A1.3

集合论, 或更一般地, 非数值数学的研究直到 19 世纪才刚刚开始. 出版于 1830 年的 G. Peacock 的《关于代数的论文》首先提到在代数中表示对象的符号可以表示非数值项. 几年后, A. De Morgan 讨论了类似的一般代数操作. G. Boole 的*Investigation of the laws of Thought*(1845) 扩展并形式化了 Peacock 和 De Morgan 的工作, 提出了集合和逻辑的形式代数. 伟大的哲学数学家 Bertrand Russell 曾写到 "是 Boole 发明了纯数学". 随后, 人们发现, 为了获得坚实的数学基础, 数值数学也需要用集合论来定义 (参见 Halmos 的《朴素集合论》). 关于集合论的详细历史, 可以参阅 Boyer 的《数学历史》的第 26 章.

练习

1. 设 A 是所有小于 30 的正整数的集合. 设 B 是所有以 7 或 2 结尾的正整数的集合. 设 C 是所有 3 的倍数的集合. 列出下列集合中的各数.

   (a) $A\cap(B\cap C)$. **(b)** $A\cap(B\cup C)$. **(c)** $A\cap(\overline{B\cup C})$. **(d)** $A-(B\cap C)$.

2. 设 $A=\{1,2,4,7,8\}$, $B=\{1,5,7,9\}$, $C=\{3,7,8,9\}$ 且 $\mathscr{U}=\{1,2,3,4,5,6,7,8,9,10\}$, 求下列集合的集合表达式.

(a) $\{7\}$. (c) $\{1\}$. (e) $\{4,8,10\}$.

(b) $\{6,10\}$. (d) $\{2,7,9\}$. (f) $\{3,5,6,7,9,10\}$.

3. 假设在例 1 中, 我们只知道这样的一条信息: 两台计算器有内存但不可充电. 现在说明我们能够推出图 A1.1 中其他 3 个盒子里的机器数量.

4. 已知图 A1.1 中的各数值, 给出挑选出 4 台机器的子集 (其中两台可充电且两台有内存) 的另外两种策略. 每一种策略有多少个选择?

5. 假设有一组 20 个人, 其中 13 名是女性. 假设女性中有 8 名已经结婚. 对于下面的每一种情况, 说明是否这条额外信息就足以确定已婚男子的人数. 如果是, 给出这个数.

(a) 有 12 个人或者已婚或者是男性 (或者同时是二者).

(b) 有 8 个人未婚.

(c) 有 15 人或者是女性或者是未婚的.

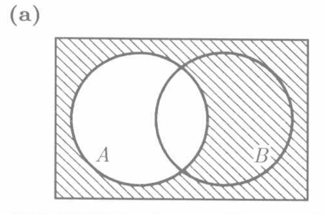
6. 对于图 A1.2a 的 Venn 图, 用每个区域所代表的集合的集合表达式来对该区域赋标签 (例如, 两个集合的交的区域将会被标签为 $A \cap B$).

7. 画出下列各集合的 Venn 图.

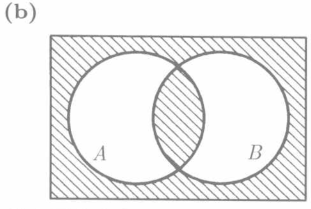
(a) $A - B$. (b) $\overline{A \cup B}$. (c) $(\overline{A \cup B}) \cap (\overline{A \cap B})$. (d) $A - (B - A)$.

8. 为下列各 Venn 图的阴影区域写出尽可能简单的集合表达式.

(a)



(b)



9. 创建表示集合 A, B, C 的 Venn 图. 这个图应该有 A, B, C 的所有可能交集的区域. 有多少不同的区域? 用每个区域所代表的集合的集合表达式标出这个区域 (例如, 3 个集合 A, B, C 的交的区域应该有标签 $A \cap B \cap C$).

10. 画出集合 A, B, C 的 Venn 图, 并把表示例 1 的集合用阴影区域表示.

11. 使用 Venn 图, 证明下面各 Boole 代数规则.

(a) $(A \cup B) \cap A = A$. (d) $(A \cup B) \cap C = (A \cap C) \cup (B \cap C)$.

(b) $(A \cap B) \cup A = A$. (e) $(A \cap B) \cup C = (A \cup C) \cap (B \cup C)$.

(c) $A \cup \overline{A} = \mathscr{U}$.

12. 使用 Venn 图, 确定下面哪些集合表达式对是等价的.

(a) $(A \cap B) \cup (\overline{A} \cap B)$. (c) $(\overline{A} - \overline{B}) \cap (\overline{\overline{A} \cup \overline{B}})$.

(b) $A - (B - (A - B))$. (d) $(\overline{A} \cup (B - \overline{A}))$.

13. 使用规则 BA2 证明 $\overline{A \cap B \cap C} = \overline{A} \cup \overline{B} \cup \overline{C}$[注意, $A \cap B \cap C = (A \cap B) \cap C = A \cap (B \cap C)$].

14. 使用 BA3 证明 $\overline{A \cup B \cup C} = \overline{A} \cap \overline{B} \cap \overline{C}$[注意, $A \cup B \cup C = (A \cup B) \cup C = A \cup (B \cup C)$].

15. 使用 BA1, BA2, BA3 以及等式 $A \cap A = A, A \cup A = A, A \cap B = B \cap A, A \cup B = B \cup A$ 来简化下面的集合表达式.

(a) $(\overline{A \cup B}) \cap (A \cap B)$. (c) $(\overline{A \cup C}) \cap (\overline{A \cup B}) \cap (\overline{B \cup C})$.

(b) $A - (\overline{B} \cup \overline{A})$. (d) $(A \cup B) \cap [(C \cup B) - (\overline{A} \cap \overline{B})]$.

16. 假设投掷两枚骰子. 至少在一枚骰子上出现 1 或 2 的结果有多少种?

17. 假设从 52 张纸牌中连续 3 次抽出一张牌 (并在下次抽取之前放回纸牌). 设 S 是第一张牌是黑桃的结果集合, H 是第二张牌是红心的结果集合, C 是第三张牌是方片的结果集合. 对于下列每一个集合 E, 用集 S, H, C 写出 E 的集合表达式, 并确定 $N(E)$.

(a) 设 E 是至少包含在 S, H, C 中的一个中的结果集合.

(b) 设 E 是至少包含在 S, H, C 中的一个中, 但是不包含在所有 3 个集合中的结果集合.

(c) 设 E 是正好包含在 S, H, C 中的两个集合中的结果集合.

A.2 数学归纳法

组合数学和计算机科学中最有用且最简单的证明技术就是**数学归纳法**(mathematical induction). 设 p_n 表示涉及 n 个对象的陈述. 于是, 对于所有的 $n \geqslant 0$, p_n 为真的归纳证明需要两个步骤.

(1) 初始步骤: 验证 p_0 为真.

(2) 归纳步骤: 证明如果 $p_0, p_1, p_2, \cdots, p_{n-1}$ 是真, 那么 p_n 一定是真.

有时候 p_n 只对 $n \geqslant k$ 成立. 这时, 初始步骤要验证 p_k 是真, 而在归纳步骤, 我们只假设 $p_k, p_{k+1}, \cdots, p_{n-1}$ 为真.

例 1 求和公式

设 s_n 表示整数 1 到 n 的和, 即, $s_n = 1+2+3+\cdots+n$. 证明 $s_n = \frac{1}{2}n(n+1)$.

使用归纳法证明这个公式. 因为 s_0 没有定义, 所以初始步骤是验证 s_1 为真. 这个公式说 $s_1 = \frac{1}{2} \times 1(1+1) = 1$, 显然是正确的. 对于归纳步骤, 假设这个公式对 $s_1, s_2, \cdots, s_{n-1}$ 为真. 在这个问题中 (如在大多数归纳问题中一样), 只需要假设这个公式对 s_{n-1} 为真.

$$s_{n-1} = 1+2+3+\cdots+(n-1) = \frac{1}{2}(n-1)[(n-1)+1] = \frac{1}{2}(n-1)n$$

现在, 我们使用 s_{n-1} 的表达式来证明这个公式对 s_n 为真.

$$\begin{aligned} s_n &= [1+2+\cdots+(n-1)]+n = s_{n-1}+n = \frac{1}{2}(n-1)n+n \\ &= \frac{1}{2}[(n-1)n+2n] = \frac{1}{2}n(n+1) \end{aligned}$$

这就完成了对于所有正整数 n, $s_n = \frac{1}{2}n(n+1)$ 的归纳证明. □

例 2 种群增长模型

假设一群蚂蚁的数量在第 2 年都变成两倍. 蚂蚁的初始数量为 10 只. 经过 n 年后这群蚂蚁将变成多少?.

设 a_n 表示 n 年后的蚂蚁数量. 我们有 $a_0 = 10$. 因为数量每年都成倍增加, 所以 $a_1 = 20$, $a_2 = 40$, $a_3 = 80$, 查看前几项 a_n 的值后, 我们得出这样的结论: $a_n = 2^n \times 10$. 对于初始步骤, 我们检验 $a_0 = 10 = 2^0 \times 10$, 符合要求. 对于归纳步骤, 假设 $a_{n-1} = 2^{n-1} \times 10$ 成立. 现在, 使用蚂蚁数每年成倍增长的性质, 得:

$$a_n = 2a_{n-1} = 2(2^{n-1} \times 10) = 2^n \times 10$$ □

例 3 素因子分解

素数是除了 1 和它自身外不能被其他正整数整除的整数 $p > 1$. 证明: 任何大于 1 的整数 n 可以写成素数之积.

使用数学归纳法证明这一事实. 因为此论断只涉及 $n > 1$, 所以初始步骤是验证 2 可以写成素数之积. 但是 2 自身就是一个素数. 所以显然 2 是素数之积 (实际上就是单个素数, 也即它本身). 接下来, 假设数 2, 3, $\cdots$, $n-1$ 可以写成素数之积, 并使用这一假设证明 n 也可以写成素数之积. 如果 n 本身是素数, 那么正如 2 的情况, 显然容易证明. 假设 n 不是素数, 那么存在整除 n 的整数 m, 且对某个整数 k, 有 $n = km$. 因为 k 和 m 都必须小于 n, 所以它们都可以写成素数之积. 把 k 和 m 的两个素数之积相乘, 得到所需的 n 的素数积表示. □

尽管希腊人在几何计算中已经使用了迭代并使用了反证法的原理, 数学归纳法最初是由 Maurolycus 大约在 1550 年左右开始使用的. 1654 年左右, 帕斯卡使用归纳法讨论验证了所谓帕斯卡三角形的数组中二项式系数和的性质 [这一性质是 5.5 节的等式 (3)]. 术语 "归纳" 是由 De Morgan 在大约 200 年后杜撰出来的. 对于数学归纳法的历史的详尽讨论, 参见 Bussey,《数学归纳法的起源》, 美国数学月刊, 1917.

练习

1. 使用归纳法证明 $1+3+\cdots+(2n+1) = (n+1)^2$.

2. 使用归纳法证明 $1^2+2^2+\cdots+n^2=\frac{n(n+1)(2n+1)}{6}$.

3. 使用归纳法证明 $-1^2+2^2-3^2+\cdots+(-1)^n n^2=(-1^n)\frac{n(n+1)}{2}$.

4. 使用归纳法证明 $(1\times 2)+(2\times 3)+\cdots+n(n+1)=\frac{n(n+1)(n+2)}{3}$.

5. 使用归纳法证明 $1^3+2^3+\cdots+n^3=\frac{n^2(n+1)^2}{4}$.

6. 使用归纳法证明 $(1+2+\cdots+n)^2=1^3+2^3+\cdots+n^3$(可运用例 1 中的结果).

7. 使用归纳法证明 $\frac{1}{1\times 2}+\frac{1}{2\times 3}+\cdots+\frac{1}{n\times(n+1)}=\frac{n}{n+1}$.

8. 使用归纳法证明 $\frac{1}{1\times 4}+\frac{1}{4\times 7}+\frac{1}{7\times 10}+\cdots+\frac{1}{(3n-2)\times(3n+1)}=\frac{n}{3n+1}$.

9. 使用归纳法证明 $(1\times 1!)+(2\times 2!)+\cdots+(n\times n!)=(n+1)!-1$.

10. 使用归纳法证明: 对于 $a\neq 1$, $\frac{1-a^{n+1}}{1-a}=1+a+a^2+\cdots+a^n$.

11. 使用归纳法证明 $1^2+3^2+5^2+\cdots+(2n-1)^2=\frac{n(2n-1)(2n+1)}{3}$.

12. 如果 Smith 博士的农场的小牛出生数量 a_n 符合规律 $a_n=3a_{n-1}-2a_{n-2}$, 而且在头两年 $a_1=3, a_2=7$, 那么证明 $a_n=2^{n+1}-1$.

13. 证明: 任意正整数有唯一的素因子分解. 可以运用例 3 的结果; 你需要使用这样的事实: 如果素数 p 能够整除若干正整数的积, 那么 p 可以整除其中的一个整数.

14. 编写一个寻找并打印前 50 个整数的素因子分解的计算机程序.

15. 使用归纳法证明: 对于任意的整数 $m>0, m\times n=n\times m$. $r\times s$ 指的是 s 的 r 个拷贝之和.

16. 使用归纳法证明: 对于整数 $n\geqslant 5$, $2^n>n^2$.

17. 使用归纳法证明: n 个对象的集合的不同子集的数量是 2^n.

18. 使用归纳法证明: $\overline{A_1\cap A_2\cap\cdots\cap A_n}=\overline{A}_1\cup\overline{A}_2\cup\cdots\cup\overline{A}_n$(参见 A.1 节练习 13).

19. 使用归纳法证明: 设投掷 n 枚不同的骰子, 各面的和是偶数的结果数等于各面的和是奇数的结果数.

20. 使用归纳法证明: 有偶数个 1 的 n 位二进制序列数等于有奇数个 1 的 n 位二进制序列数.

21. 使用归纳法证明: 3 个连续正整数的立方和可以被 9 整除.

22. 某郊外住宅区的猫的数量有这样的性质: 每年的猫的数量等于前两年的猫的数量和. 如果第一年有 1 只猫, 而第二年有 2 只猫, 那么使用归纳法证明: 第 n 年的猫的数量等于

$$\frac{1}{\sqrt{5}}\left[\left(\frac{1}{2}+\frac{1}{2}\sqrt{5}\right)^{n+1}-\left(\frac{1}{2}-\frac{1}{2}\sqrt{5}\right)^{n+1}\right]$$

23. 为什么不能使用归纳法证明所有有限长度的二进制序列的个数是有限的?

24. 下面 n 元集合中所有元素 $x_1,\cdots,x_n$ 相等的归纳证明中有什么错误?

(a) 初始步骤 $(n=1)$: 对于只有一个元素 x_1 的集合, x_1 等于它自身.

(b) 归纳步骤: 假设 $x_1=x_2=\cdots=x_{n-1}$. 因为根据归纳假设 $x_{n-1}=x_n$(此时, 把 x_{n-1}, x_n 考虑成一个 2 元集合), 所以有 $x_1=x_2=\cdots=x_{n-1}=x_n$.

25. 下面对于 $a\neq 0$, $a^n=1$ 的归纳证明中有什么错误?

(a) 初始步骤 $(n=0)$: $a^0=1$ 永远为真.

(b) 归纳步骤: 假设 $a^{n-1}=1$, 那么 $a^n=a^{n-1}a^{n-1}/a^{n-2}=1$.

A.3 概率简介

历史上, 计数问题一直与概率紧密相关. 实际上, 任意形如 "······ 的小矮人有多少个" 这样的问题都与 "······ 的小矮人的比例是多少" 这样的形式密切相关, 而这样的问题也可以用概率的形式提出来 "随机选取 ······ 的小矮人的概率是多少? "1650 年左右, 帕斯卡在研究某些博弈概率的分析过程中发现了著名的二项式系数的帕斯卡三角形表示. 投掷一枚硬币 10 次得到至少 7 个正面的概率, 从一副洗好的纸牌中分

配到有一对或更好的 5 张手牌的概率, 如果 20 台样本计算器中 5%的样本机器是次品, 求得到一台次品机器的概率, 以上这些概率实际上都是计数问题.

200 年前, 法国数学家拉普拉斯首先如下定义了概率:

$$概率 = 希望事件的数量/事件总量$$

这个定义与"大街上的人"对概率的直观想法相符合. 本文仅处理运用拉普拉斯概率定义的概率问题. 这个定义中隐含的假设是每一个事件有等同的可能性. 如果事件总数是无限的, 例如, 0 与 1 之间的所有实数, 那么我们就不能运用拉普拉斯的定义. 一个更棘手的困难是在某些概率问题中, 必须小心定义"事件", 否则它们不可能有等同的可能性, 例如, 当投掷一枚硬币 10 次时, 看到的可能的正面个数. 为了阐明这一困难, 我们必须引入概率论的一些基本的术语.

实验(experiment) 是明确定义的过程, 它产生给定结果集合中的一个结果. 这些结果被称为**基本事件**(elementary event), 而所有基本事件的集合称为这个实验的**样本空间**(sample space). 我们只对基本事件有相等可能性的实验感兴趣. 作为若干基本事件的子集的事件称为**复合事件**(compound event).

例如, 当投掷一枚骰子时, 那么得到一个特定数, 比如 5, 这就是一个基本事件, 而得到偶数则是复合事件. 如果 S 是实验的样本空间, 而且 E 是 S 中的事件, 那么拉普拉斯的概率定义说 E 的概率 $\text{prob}(E)$ 是:

$$\text{prob}(E) = N(E)/N(S)$$

在大多数情况下, S 的大小一般很容易确定, 所以确定 $\text{prob}(E)$ 的问题就简化成为计数在事件 E 中的结果 (基本事件) 数. 回到投掷骰子的问题, 结果的样本空间是 $S = \{1, 2, 3, 4, 5, 6\}$. 如果事件 E 是得到一个偶数, 那么 $E = \{2, 4, 6\}$, $\text{prob}(E) = 3/6 = 1/2$.

我们讨论的很多实验是一系列重复的简单实验. 每一轮简单实验被称为一个试验 (trial). 例如, 投掷一枚硬币 3 次的实验包含投掷一枚硬币一次的简单实验的 3 次连续试验. 投掷两枚骰子并记录投掷出的两个值的和包含同时投掷一枚骰子的两个简单实验. 在包含多次试验的任意实验中, 基本事件是简单实验的结果序列. 如果是同时进行简单实验, 我们对这些简单实验进行标号, 并以实验标号的顺序列出它们的结果.

投掷一枚硬币 3 次的实验的基本事件样本空间是

$$\{\text{HHH, HHT, HTH, HTT, THH, THT, TTH, TTT}\} \tag{1}$$

投掷两枚骰子的实验的样本空间是:

$$S = \{(i, j) \mid 1 \leqslant i \leqslant 6, 1 \leqslant j \leqslant 6\}$$

仔细考虑投掷两枚骰子的实验, 并记录两枚骰子上的两个值的和 (样本空间就是上面给出的集合 S).

根据事件的形式定义, 两个值的和等于 k 不是基本事件, 而是复合事件. 两枚骰子的和等于 7 的事件是基本事件的子集:

$$S_1 = \{(1, 6), (2, 5), (3, 4), (4, 3), (5, 2), (6, 1)\}$$

因此, prob(和 =7)$=N(S_7)/N(S) = 6/36$. 类似地, $S_2 = \{(1, 1)\}$, 所以 prob(和 =2)$= N(S_2)/N(S) = 1/36$. 观察发现, 和为 7 的概率是和为 2 的概率的 6 倍. 如果我们是把两枚骰子的可能和考虑成基本事件, 那么我们应该有样本空间 $S^* = \{2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12\}$, 而且每个和应该有概率 $1/N(S^*) = 1/11$, 这显然是错误的. 如果把正面的数量考虑成为投掷 3 枚硬币的实验中的基本事件 (参见 (1) 所列出的正确的样本空间), 我们就会犯同样的错误.

假设有一个盒子, 里面有 5 个相同的红球和 20 个相同的黑球. 我们的实验是取出一个球. 基本事件的样本空间是什么? 作为结果, 我们能够记录的只能是这个球的颜色, 这样导致样本空间 S={红, 黑}. 于是, 根据拉普拉斯的概率定义, 挑出红球的概率是 1/2. 尽管这个实验只包含一个试验, 显然我们犯了把骰子的和处理成为基本事件时所犯的同类错误. 在阅读下面的内容之前, 我们希望读者停顿一下, 并试着对解决红–黑球悖论给出说明.

合理的解决方案是不考虑从这个盒子里抽出什么颜色的球, 而是我们直接把手伸进这个盒子里抓出一个球. 当球在盒子里是按位置放置时, 有 25 个不同的空间位置. 所以在这种意义下, 有 25 种不同的事件. 我们用下面的规则解决这种复杂的"相同"对象的问题.

相同对象规则

在概率问题中, 没有相同对象集合, 所有对象都是可区分的.

对于复杂实验, 我们总是应该花一点时间来确定基本事件是否可区分的.

概率研究的原始动机就是本书中的机会游戏的概率研究, 例如, 投掷骰子, 投掷硬币, 纸牌游戏等. 1500 年代, Cardano 计算了两枚骰子的不同和的概率. 1600 年左右, 伽利略 (Galileo) 计算了 3 枚骰子不同和的概率. 1650 年左右, 帕斯卡与费马之间进行了一系列关于博弈概率的书信来往, 这些书信成为概率论的真正开始. Jacques Bernoullli 的《猜度术》(1713) 第一次对概率 (及相关的组合方法) 做了系统处理. 一百年后, 拉普拉斯出版了他的伟大著作《概率分析理论》, 此书既包含本书所使用的有限样本空间的概率定义, 又包含了以现代微积分为基础的现代概率论推演. 参见 F. N. David, 《游戏、上帝与赌博: 概率的历史》(Dover, 1998), 此书对概率论的历史做了很好的阐述.

练习

1. 随机选取 5~12 之间的一个整数 (包括 5 和 12). 这个数是偶数的概率是多少?

2. 在投掷一枚硬币 3 次的实验中, 设 E_k 是正面个数等于 k 的复合事件. 确定 $\text{prob}(E_k)$, 其中 $k = 0, 1, 2, 3$.

3. 在投掷两枚不同骰子的实验中, 求下面事件的概率.

(a) 两枚骰子显现出相同的值. **(b)** 骰子的和是偶数.

(c) 骰子的和是其中一个骰子的值的平方.

4. 投掷一枚骰子 3 次. 5 至少出现两次的概率是多少?

5. 随机选取字母 h, a, t 的一个排列, 求下面事件发生的概率.

(a) 这些字母以字母表的顺序出现.

(b) 在一个排列中, 字母 h 出现在字母 t 的后面.

6. 随机选取集合 1, 2, 3, 4, 5 中的两个数字的子集, 求下面的事件发生的概率.

(a) 这个子集是{1, 2}. **(b)** 1 不在这个子集中.

7. 一个缸里有 6 个红球和 3 个黑球. 如果选出一个球, 然后再放回, 再选出第二个球, 下面各事件发生的概率是多少?

(a) 两个球都是黑球. **(b)** 一个球是黑球而另一个球是红球.

8. 一个缸里有两个黑球和 3 个红球. 如果连续拿出两个不同的球, 那么这两个球有相同颜色的概率是多少?

9. 把两个男孩和两个女孩随机排成一行. 女孩和男孩交替出现的概率是多少?

10. 投掷 5 枚不同的骰子. 至少得到一个 6 的概率是多少?

11. 设一所学校有 100 名学生, 其中 50 人选取了法语, 40 人选取西班牙语, 20 人选取了法语和西班牙语, 那么随机选取一名学生, 他没有选取任何语言的概率是多少?

12. 在 1~50 之间且不等于 1 和 50 的整数可以被 3 或 4 整除的概率是多少?

13. 有 3 个缸, 每个缸里面有两个球: 第一个缸里有两个黑球, 第二个缸里有两个红球, 第三个缸里有一个红球和一个黑球. 如果随机选出一个缸, 并随机从这个缸里拿出一个球, 这个球是红球, 那么这个缸里的另一个球也是红球的概率是多少? (提示: 设样本空间为第一次选出的是红球的所有实验结果的集合.)

14. 下面问题的样本空间是什么: 一个缸有 10 个黑球和 5 个红球; 拿出 4 个球, 但是不能看到(我们闭着眼睛), 然后再拿出第 5 个球并看其颜色. 此时这第 5 个球是红球的概率是多少?

15. 在下面每一次实验中, 把什么作为基本事件才能使得它们有相同的机率?

(a) 投掷一枚硬币直到出现两次正面, 或者直到这枚硬币被投掷 10 次.

(b) 随机选出一个正整数.

(c) 随机选出两个正整数, 并记录它们的差.

(d) 从有两个红球和两个黑球的缸里随机选出一个球, 然后再把这个球放回, 再选出一个球, 直到得到一个红球为止.

A.4 鸽巢原理

鸽巢原理是最简单的形象化思维方式的一种，而且它的一般形式包含整个组合理论中最重要且最困难的一些结果. 把这一课题安排在附录中的原因是它与本书的任意一章都不能自然地融合在一起. 大多数关于组合数学的图书都用一章的篇幅来研究拉姆齐理论，例如，参见 Brualdi 的《组合数学导论》(Prentice Hall, 2004). 对于拉姆齐理论的详尽研究，参阅 Graham, Rothschild, and Spencer 的《拉姆齐理论》(John Wiley, 1990).

鸽巢原理

如果鸽子数量比鸽巢多，那么某个鸽巢里必定包含两只或更多的鸽子. 更一般地，如果鸽子多于鸽巢的 k 倍，那么某个鸽巢必定至少有 $k+1$ 只鸽子.

这一原理也称为狄利克雷抽屉原理. 这一原理的一个应用就是发现纽约城至少有两个人的头上有相同数量的头发. 纽约城有超过 7 300 000 个人，平均头皮上所含头发数是 100 000 根. 实际上，鸽巢原理使我们可以断言，一节地铁列车运载的大约 73 个人的头上理论上有可能有相同数量的头发.

下面 3 个问题给出鸽巢原理的不同扩展形式.

(1) 1 到 200 之间的不同整数的集合需要包含多少个整数，才能断言这个集合中存在两个有公因子的数？

(2) 1 到 n 之间的不同整数的集合需要包含多少个数，才能断言这个集合包含 5 个间隔相等的整数 a_1, a_2, a_3, a_4, a_5 的子集，即 $a_2-a_1=a_3-a_2=a_4-a_3=a_5-a_4$？

(3) 给定一个正整数 k，一群人中有多少个人才能断言在这一群人中存在 k 个人的子集，他们互相认识，或者存在 k 个人的子集，他们互相不认识？

对于问题 3，有可能有这样的情况：对于充分大的 k，存在任意多的一组人，他们不满足 k 个人相互是朋友或 k 个人相互不认识这一性质. 这不成立. 拉姆齐理论的一个定理说的是：存在有限大小的数 r_k，使得至少有 r_k 个人的任意组合都满足这个性质. 这个定理并没有说数 r_k 是多少. 实际上，我们只对较小的 k 知道 r_k 的值.

大多数鸽巢原理的扩展形式需要特殊的、具有一定水准的技巧，而不是本书所探讨的组合求解问题的逻辑和技术. 然而，我们还是在本书中引入这个基本的鸽巢原理，因为有时候它很有用. 下面的例子中，一个例子说明这一原理的直接使用，而另外两个例子说明其间接应用.

例 1 汇集回应

得克萨斯的西部地区有 20 个小镇，我们想从这些小镇之一选出 3 个人来帮助调查它们的城镇. 如果进入任意特定的城镇，并发布广告寻求帮手，由以往的经验我们知道得到 3 个回应者的机会很少. 相反，我们在发送到所有 20 个城镇的地方报纸上发布广告. 我们需要多少回应者才能保证回应者集合包含 3 个来自同一个小镇的人呢？

根据鸽巢原理，我们需要多于 $2\times 20=40$ 名的回应者. □

例 2 为计算机连接打印机

我们有 15 台小型计算机和 10 台打印机. 每 5 分钟，若干台计算机需要打印机. 计算机和打印机之间需要多少不同的连接，才能确保如果 10 台 (或更少) 计算机需要打印机时，总是存在使得每台计算机都使用不同的打印机的连接？

使用鸽巢原理的变形，我们知道至少要存在 60 种连接. 否则 10 台打印机中有一台打印机，比如说是 A 就会与 5 台 (或更少) 计算机连接，而且与 A 连接的 5 台计算机中的任意一台都不是寻求打印机的 10 台计算机中的一台，那么这 10 台计算机中的任意一台都不使用打印机 A. 如果设计得当，恰好 60 种连接将解决连接问题 (参见练习 15). □

例 3 和等于 4 的子集

证明和等于 20 的 8 个正整数的任意集合中有和等于 4 的子集.

我们证明这个集合一定有下面和等于 4 的四个子集中的一个：(a) 四个 1；(b) 两个 2；(c) 两个 1 和一个 2；(d) 一个 1 和一个 3. 这些可能性之一一定成立是鸽巢类型的讨论.

如果 S 不包含 1 和 2, 那么它的和至少应该是 $3\times 8=24$. 所以根据鸽巢原理, S 必须包含一个 1 或一个 2.

假设 S 包含一个 2. 如果有另一个 2 或者如果有两个 1, 那么我们就完成了证明. 可能有一个 1, 但是所有其他整数都至少是 3. 如果没有 1, 其他 7 个整数都至少是 3, 且它们的和一定等于 $20-2=18$: 根据鸽巢原理类型的讨论, 这是不可能的. 如果存在一个 1, 其他 6 个整数都至少是 3, 且它们的和一定等于 $20-2-1=17$: 这也是不可能的.

假设 S 至少包含一个 1 但是不包含 2. 如果 S 中有一个 3, 那么可以运用情况 (d). 如果 S 中有四个 1, 则可以运用情况 (a). 可选择的是 S 中没有 2 也没有 3, 且至多有三个 1. 那么在 S 中至少有 5 个大小至少为 4 的整数, 它们的和又必须小于 20, 这是不可能的. □

练习

1. 给定一组 n 名女子和她们的丈夫, 从这群 $2n$ 个人中必须选出多少个人才能确保这个集合包含一对夫妇?

2. 证明: 在一个有 20 个人参加的聚会上, 一定有 2 个人 (在聚会中) 有相同数量的朋友.

3. 在一次循环赛中, 证明如果没有选手输掉所有的比赛, 那么一定有 2 名获胜场次数相同的选手.

4. 给定 10 本法语书、20 本西班牙语书、8 本德语书、15 本俄语书和 25 本意大利语书, 必须选出多少本书才能确保其中有 12 本书是同种语言的书?

5. 设在一次有 20 个人参加的聚会上, 有 48 对互相认识的人, 证明有一个人有 4 个或更少熟人.

6. 一名教授每年在她的伦理课上讲 3 个笑话. 这位教授需要有多少个笑话才能确保 12 年中任意两年的笑话都不完全相同?

7. 给定 7 个不同整数的任意集合, 证明在这个集合中一定存在两个整数, 它们的和或差是 10 的倍数.

8. 证明: 如果从 $1,2,\cdots,2n$ 中选出 $n+1$ 个不同的数, 那么这 $n+1$ 个数中存在两个相邻的数.

9. 假设把 1 到 10 的数沿着一个圆随机放置. 证明某 3 个相邻数的和至少为 17.

10. 一台计算机在为期 12 天内总共被使用了 99 个小时, 每天使用的小时数是整数. 证明在某连续 2 天内, 这台计算机至少使用了 17 小时.

11. 证明: 1 到 14 之间的 8 个不同整数的任意子集包含一对整数 k 和 l, 使得 l 整除 k.

12. 证明: 在大于等于 3 的 n 个整数的任意集合中, 总是存在一对整数, 它们的差可以被 $n-1$ 整除.

13. 证明: 2 到 $2n$ 之间的 $n+1(n\geqslant 2)$ 个不同整数的任意子集总包含一对没有公因子的整数.

14. 证明: 任意一组和等于 30 的 16 个 (不全部不同的) 正整数中, 有一个和等于 n 的子集, 其中 $n=1,2,\cdots,29$.

15. 在例 2 中, 寻找所需的 60 种计算机–打印机连接的集合.

16. 一个大型计算中心曾经有 6 台机算机和 10 台打印机. 每台计算机都与其中若干台打印机连接. 现在, 10 台旧打印机被 6 台性能更可靠的打印机取代, 但是, 暂时还允许计算机相信有 10 台打印机. 4 台虚拟打印机将把计算机的需求传递给另 6 台真实的打印机. 4 台虚拟打印机与 6 台真实打印机需要多少连接才能处理任意 6 个打印需求?

17. 证明: 对于 1 到 60 之间的 10 个不同整数的任意集合 S, 总存在 S 的两个不相交子集 (不必用尽 S 中的所有数), 它们的和相等.

18. 两个圆盘都有 10 个 0 和 10 个 1 以不同的顺序等间隔地分布在它们的边上. 证明我们总是能够把其中一个圆盘放在另外一个圆盘之上, 使得至少有 10 个位置有相同的数字.

19. 一名学生 7 周内每天至少花 1 个小时学习编织, 但是每周都不超过 11 个小时. 证明总是存在连续的几天, 其间这个学生总共正好学习了 20 个小时.

20. 如果 G 是一个 n 顶点图, 其中每个顶点的度 $\geqslant (n-1)/2$, 证明 G 是连通的, 即, 存在连接 G 中每一对顶点的路径.

21. 证明: n^2+1 个不同数的任意序列包含一个 $n+1$ 个不同数的递增子序列, 或 $n+1$ 个数的递减子序列.

22. 证明: 在至少有 6 个人的聚会上, 总是存在相互是朋友的 3 个人, 或者相互不认识的 3 个人.

A.5 计算复杂度和 NP 完备性

本节概括计算复杂度的基本思想. 毫不奇怪, 计算机科学重点关心的一个问题是计算速度, 无论这一计算是探测太空船轨道的数值算法, 还是调度航线或者调度数据库的信息遍历搜索的运筹学程序. 对于某些计算问题, 如 (以字母表顺序) 排序 n 个单词的列表, 我们已经知道一些速度较快的程序. 而对其他的一些问题, 例如, 旅行商问题 (确定每条边都有长度的 n 顶点图的最短哈密顿回路) 这样的计算对于适度的 n 都可能花费难以置信的大量时间, 比如 $n = 100$.

描述**计算复杂度**(computational complexity), 或速度的标准方法是使用一个函数 $f(n)$, 这个函数的值是执行这个程序所需的最大的计算机步数 (例如, 加法、乘法、比较), 其中 n 是输入数据的 "大小". 这个大小可以是图的顶点数量, 也可以是要排序的项的数量. 最常见的函数是线性函数. 当输入的大小变大时, 计算时间将按一定的比率增加.

更快的函数是对数函数. 使用适当的搜索树, 正如在 3.1 节所讨论的那样, 可以用 $\log_2 n$ 个步骤 (比较) 区分出 n 个单词的词典中的未知单词. 其关键是可以重复地比较这个未知单词是在测试单词的前面, 还是在它的后面, 使得每一次比较都把可能的答案减少一半. 可能的答案是二叉树的叶子, 内部顶点是测试单词. 这个搜索的复杂度是树高. 有 n 个叶子的平衡二叉树的高度是 $\log_2 n$.

比线性函数稍慢的函数是形如 $An\log_2 n$ 的函数, 其中 A 是某个常数. 这是 3.4 节给出的排序 n 个项的堆排序算法的比较次数. n 个项的列表有 $n!$ 种可能的顺序. 为了仅以 $An\log_2 n$ 次比较在 $n!$ 种可能的答案做出选择需要一种算法, 就像求未知单词的 $\log_2 n$ 的识别算法一样, 这种算法重复地把可能的顺序集合减少一半. 注意, $\log_2(n!)$ 以 $n\log_2 n$ 为界.

效率更差一点的算法的计算复杂度是关于 n 的多项式函数. 回想一下, 当 n 加倍时, 平方函数以 4 的倍数增长, 而立方函数则以 8 的倍数增长, 当 n 加倍时, k 阶函数以 2^k 的倍数增长. 以现代计算机的速度, 特别是并行超级计算机, 复杂度是 n^4 和 n^5 的算法可以解决适当大小的问题, 例如 $n = 500$. 回想一下, 我们说计算复杂度函数是检查算法在最坏情况下的行为. 最坏情况的复杂度为 n^4 的算法对一些典型问题来说也许已经很快了.

比上面坏得多的计算复杂度是诸如 2^n 或者 $n!$ 这样的指数函数. 对于 $n!$ 的复杂度, n 只简单地增加 1, 比如说从 49 到 50, 计算量就增加 50 倍. 旅行商问题的所有已知算法都有指数复杂度. 计算机科学家尝试着确定解决某个问题的计算复杂度的下界. 而其中特别感兴趣的是对诸如旅行商问题这样非常困难的问题是否存在多项式算法. Cook为讨论这些困难问题研发了有用的框架. 他以可满足性问题为起点, 即, 确定有 n 个变量的逻辑命题是否可满足的. 我们还不知道是否有比评估 n 变量逻辑表达式的所有可能的 2^n 个真假值更快的算法 (关于逻辑命题的有关背景参见附录 A.1). 于是, Cook 研究了使用其他难题对可满足性问题的建模, 并反过来把其他难题建模成可满足性问题.

一个问题称为是**NP完备**(NP-complete) 的, 如果这个问题可以对可满足性问题建模, 而且它可以建模成可满足性问题. 建模所需的时间必须是多项式时间. 这些模型通常很复杂. 例 1 给出了一个比较简单的建模过程, 在例 1 的后面我们展示如何把确定一个图是否可以顶点 3 着色的问题建模成一个可满足性问题. 对于 NP 完备模型的更全面的研究, 参阅任何算法分析的图书. 注意, NP 完备的形式定义更加复杂, 但这里给出的定义是等价的.

NP 完备问题是需要回答 "是" 和 "否" 的问题. 例如, 任意 n 顶点图是否可以用 3 种颜色着色的问题是 NP 完备问题, 一个图是否可以用大于 2 的任意指定数量的颜色着色的问题也是 NP 完备问题. 为了求 n 顶点图的色数 (最少颜色数), 我们做二分搜索, 看这个图是否 $n/2$ 可着色的; 如果是, 它是否 $n/4$ 可着色的; 如果不是, 它是否 $3n/8$ 可着色的, 等等, 直到求出可以着色这个图的最小颜色数. 这一策略需要回答 $\log_2 n$ 个着色问题.

观察发现, 任意 NP 完备问题都可以经由可满足性问题这样的中间模型为其他任意 NP 完备问题建模. 这一等价性的结果是, 如果我们可以为这些 NP 完备问题中的任意一个问题找到一个多项式复杂度的算法, 那么可以利用它为所有 NP 完备问题提供多项式算法.

一般认为我们无法为 NP 完备问题找到多项式算法. 数以千计的极难问题已被证明是 NP 完备的. 在众多已知是 NP 完备问题的图问题中包括哈密顿回路和哈密顿路径的存在性、图的可着色性, 以及最大完

全子图的大小等问题. 目前我们仍不知道确定两个图是否同构的问题是不是一个 NP 完备问题.

例 1 把图的 3 着色建模成可满足性问题

把确定一个 n 顶点图 G 是否可以 3 着色的问题重新公式化成可满足性问题. 我们将创建一组简单的析取式 (用逻辑 "或" 描述的命题). 可满足性问题将确定是否存在使所有析取式为真的逻辑变量值 (真或假) 选择. 值的个数必须是 n 的多项式函数. 对于此例, 我们将使用 $3n$ 个变量.

我们在图着色中所做的选择是第 i 个顶点应该是哪种颜色: 它是颜色 1, 颜色 2, 还是颜色 3? 我们为这些可能性定义 3 个逻辑变量: $x_{i,1}$, $x_{i,2}$, $x_{i,3}$, $x_{i,j}$ = 真, 如果第 i 个顶点有颜色 j, 否则 = 假.

首先, 我们开发表示每一个顶点正好有一种颜色的限制的析取式, 即, $x_{i,1}$, $x_{i,2}$, $x_{i,3}$ 中正好一个为真. 下面的复合命题表示这一限制.

$$(x_{i,1} \wedge \overline{x}_{i,2} \wedge \overline{x}_{i,3}) \vee (\overline{x}_{i,1} \wedge x_{i,2} \vee \overline{x}_{i,3}) \vee (\overline{x}_{i,1} \wedge \overline{x}_{i,2} \wedge x_{i,3}), \quad i = 1, 2, \cdots n,$$

通过对析取式反复使用分配律, 并消除多余的项, 我们得到一组与上面复合命题等价的析取式.

$$x_{i,1} \vee x_{i,2} \vee x_{i,3}, \qquad \overline{x}_{i,1} \vee \overline{x}_{i,2}, \qquad \overline{x}_{i,1} \vee \overline{x}_{i,3}, \qquad \overline{x}_{i,2} \vee \overline{x}_{i,3}, \quad i = 1, 2, \cdots, n \tag{1}$$

接下来, 我们需要表示着色的限制, 即对于 G 的每一条边 (v_i, v_j), v_i 和 v_j 不能有相同的颜色, 即, v_i, v_j 中至少有一个不是颜色 k.

$$\overline{x}_{i,k} \vee \overline{x}_{j,k}, \qquad k = 1, 2, 3, \qquad (v_i, v_j) \text{ 是 } G \text{ 中的边} \tag{2}$$

(1) 和 (2) 的简单析取命题组的可满足性问题就是 G 是否可以 3 着色这个问题的命题形式. □

关于计数和图论的术语表

排列 排列是对象的序列，或有序列表. r 排列是 r 个对象的排列. 排列可能允许也可能不允许对象重复. 从 n 个对象中没有重复地选出 r 个对象的 r 排列数是 $P(n,r)$. 从 n 个类型的对象中可重复地选出 r 个对象的 r 排列数是 n^r.

二项式系数 二项式系数是诸如 $(a+x)^n$ 这样的多项式的二项式展开式中的系数. x^r 在 $(1+x)^n$ 中的系数记作 $C(n,r)$，或 $\binom{n}{r}$. 这一系数等于 n 元集合的不同 r 元子集的数量.

二部图 二部图 $G=(X,Y,E)$ 是一个图，其顶点被分割成为两个顶点集合 X 和 Y，而 G 中的每条边连接 X 中的一个顶点和 Y 中的一个顶点.

色数 一个图的色数是着色该图时的最少可能颜色数. 参见*着色图*.

色多项式 色多项式 $P_k(G)$ 告诉我们有多少种方法 k 着色图 G 的顶点.

回路 回路是顶点 $(x_1, x_2, x_3, \cdots, x_n)$ 的序列，其中 $x_1=x_n$，x_i 与 x_{i+1} 相邻. 一个顶点在回路中只能出现一次 (作为起始顶点和结束顶点的相同顶点除外).

着色图 着色一个图 G 是给每一个顶点分配一种颜色，使得相邻顶点有不同的颜色. 还可以着色图 G 的每一条边，使得具有公共顶点的边有不同的颜色.

组合 一个组合是对象的一个子集，或等价地，一组无序对象. 在组合中，对象可以重复也可以不重复. 从 n 个对象中取 r 个不重复对象的不同组合数是 $C(n,r)$. 从 n 个类型的对象中重复地选出 r 个对象的不同组合数是 $C(n+r-1,r)$.

完全图 $\boldsymbol{K_n}$ 和完全二部图 $\boldsymbol{K_{m,n}}$ K_n 是 n 个顶点上的图，其中每一对顶点间都存在一条连接它们的边. K_3 是一个三角形. K_2 是一条边. $K_{m,n}$ 是一个二部图，它的两个顶点集合中分别有 m 个顶点和 n 个顶点，且所有可能边都连接两个顶点集合中的顶点.

补图或补 图 $G=(V,E)$ 的补图 $\overline{G}=(V,\overline{E})$ 与 G 有相同的顶点集合 V. 一对顶点由 $\overline{G}$ 的边相连，当且仅当它们不由 G 中的边相连.

连通分量 非连通图 G 是由一组连通分量，或“连通片”组成的. 连通图是由单一连通分量组成的. 形式上，G 的连通分量是由某个特殊的顶点 x 及从 x 出发通过 G 中的路径能够达到的所有顶点组成的.

连通图 一个图是连通的，如果任意给定的一对顶点间都存在连接它们的路径. 有向图是连通的，如果把它当作无向图时 (忽略所有边的方向) 它是连通的.

圈 圈是依次相连的边的序列，它的起始顶点与结束顶点相同，而且每条边只出现一次. 与回路不同，在圈中，顶点可以多次被访问.

乱序 不同对象的给定排列的乱序是一个重新排列，使得每个对象都不在其原来排列中的位置上.

有向图，有向边 一个图是有向图，如果每条边 $(a^{\rightarrow}, b)$ 都是有向的，从 a 到 b. 有向图可以包含诸如 $(a^{\rightarrow}, b)$ 和 $(b^{\rightarrow}, a)$ 这样的两条连接两个顶点的反向有向边.

分配 分配是从给定一组相同或不同对象到不同目标集合的赋值. 除非明确禁止，否则多个对象可以得到相同的目标值.

边覆盖 图的边覆盖是顶点的集合 S，使得图中每一条边都与 S 中的一个顶点相连.

欧拉圈，欧拉迹 图的欧拉圈 (欧拉迹) 是包含图中所有边的圈 (迹). 另外，它必定访问每一个顶点至少一次.

生成函数 对 n 个对象完成某种操作的方法数 a_n 的生成函数 $g(x)$ 是有展开式 $g(x)=a_0+a_1x+a_2x^2+\cdots+a_nx^n+\cdots$ 的多项式或幂级数. 这样的生成函数也称为普通生成函数，而相对应的指数生成函数 $g(x)$ 有 $g(x)=a_0+a_1x/1!+a_2x_2/2!+\cdots+a_nx^n/n!+\cdots$ 的形式.

图 图 $G=(V,E)$ 是由顶点的有穷集合 V 和边的有穷集合 E 组成的. 边 $e=(a,b)$ 连接两个不同的顶点 a 和 $b(a\neq b)$. 两条边不能连接同一对顶点. 除非 G 是有向图 (参见*有向图*)，(a,b) 和 (b,a) 是同一条边 (不考虑顺序).

哈密顿回路, 哈密顿路径　一个图的哈密顿回路 (路径) 是包含该图每一个顶点的回路 (路径).

独立集合　一个图的一个顶点集合是独立的, 如果其中的所有顶点对都是不相邻的.

同构, 同构图　两个图 G_1 和 G_2 是同构的, 如果 G_1 中的顶点与 G_2 中的顶点间存在称为同构的匹配, 使得 G_1 中的两个顶点相邻当且仅当 G_2 中对应的顶点相邻. 通俗地说, 两个图是同构的, 如果除了顶点的名字不同外, 它们是 "相同的图".

地图着色　给每一个国家分配一种颜色来正确地为一幅国家地图着色, 使得有共同边界的国家有不同的颜色.

匹配　二部图 $G = (X, Y, E)$ 的一个匹配是以一对一的方式把 X 中的某些顶点与 Y 中的某些顶点配对的边的一个子集.

多重图　多重图是一个扩展图, 其中 (1) 允许有多条边, 即两个顶点间可以有两条或更多条连接它们的边; (2) 允许有环, 即形如 (a, a) 的边.

网络　网络是图中每条边 e 分配了一个正整数 $k(e)$ 的图, 通常是有向图.

网络流　网络流是满足 4.3 节开头所列出的限制的网络的边上的函数.

分拆　一组相同对象的分拆把对象分成不同大小的各组. 而一个整数 n 的分拆是和等于 n 的一组正整数.

路径　路径是满足 x_i 与 x_{i+1} 相邻的顶点 $(x_1, x_2, x_3, \cdots, x_n)$ 的序列. 一个顶点在一条路径上只出现一次, 例外是有可能 $x_1 = x_n$.

置换　对象集合或序列的置换是这个集合或序列的排列, 一般不允许重复. r 置换是从这个集合或序列中选出的 r 个对象的 r 排列.

可平面图　一个图是可平面的, 如果存在一种方法可以在一张纸上画出这个图而没有边相交. 平面图 (plane graph) 是在边不相交的情况下画出的可平面图.

递推关系　递推关系是诸如 $a_n = 2a_{n-1} + 3a_{n-2}$ 这样的方程, 其中, a_n 是对 n 个对象完成某种操作的方法数, 它可以被其他的 a_k 表示, 其中 $k < n$.

选取　选取是一组无序对象. 在选取中对象可能重复也可能不重复. 从 n 个对象中不重复地选出 k 个对象的不同选取数是 $C(n, k)$. 从 n 个类型的对象中重复地选出 k 个对象的不同的选取数是 $C(n + k - 1, k)$.

子图　子图是包含在另一个图中的图. 如果 $G' = (V', E')$ 是 $G = (V, E)$ 的子图, 那么 $V' \subseteq V$, 且 $E' \subseteq E$.

迹　迹是依次相连的边的序列, 其中, 每条边只出现一次. 与路径不同, 在迹中顶点可以被访问多次.

树　树是特殊的图, 它有一个称为根的特殊顶点, 且对于每一个非根的顶点 x, 存在唯一一条从根到 x 的路径. 可以把无向树刻画为无回路连通图. 关于与树相关的术语, 参见下文.

关于树的术语表

祖先　顶点 x 的祖先是从根到这个顶点 x 的路径上的一组顶点.

回溯搜索　参见*深度优先搜索*.

平衡树　所有叶子都在层次 h 和层次 $h-1$ 上的树, 其中 h 是这棵树的高度.

二叉树　每个内部顶点正好有两个孩子的树.

广度优先搜索　从根开始, 寻找与根相邻的所有顶点 z, 然后再求这些顶点中每个顶点的所有相邻顶点, 以此类推.

孩子　顶点 x 的孩子是有从 x 出发的边 $(x\rightharpoonup, y)$ 上的顶点 y.

深度优先搜索　也称*回溯搜索*, 从根开始尽可能远地构建一条路径; 当这条路径不能延伸到未访问过的顶点时, 从当前的顶点回溯, 沿当前路径返回直到发现一个可以构建到达新顶点的旁路的顶点, 并从该顶点构建新路径.

子孙　顶点 z 是顶点 x 的子孙, 如果 x 在从根到 z 的路径上.

高度　树中的最大层次数.

中序遍历　在二叉树中, 从根开始, 这一遍历递归地列出顶点 x 的左子树上的顶点, 然后是 x, 然后再列出 x 的右子树上的顶点.

内部顶点　有孩子的顶点; 内部顶点和双亲是等价的术语.

叶子　没有孩子的顶点; 叶子的唯一一条关联边来自它的双亲.

层次或者层次数　从根到给定顶点 x 的路径长度; 例如, 一棵树的层次 2 是由从根开始长度为 2 的所有顶点组成的.

m 元树　所有内部顶点都有 m 个孩子的树.

双亲　顶点 x 的双亲是唯一的顶点 z, 使得从 z 到 x 存在边 $(z\rightharpoonup, x)$.

后序遍历　按深度优先搜索树时以最后遇到的顶点的顺序列出顶点.

前序遍历　按深度优先搜索树时以最先遇到的顶点的顺序列出顶点.

根　树的特定顶点, 它到任意其他指定顶点有唯一一条路径; 在有序树或有根树中, 根是唯一没有双亲且处于层次 0 上的顶点.

根树　所有边都是离开根的有向边的有向树.

兄弟　顶点 x 的兄弟是与 x 有相同双亲的那些顶点.

生成树　连通图的子图且包含这个图的所有顶点的树.

子树　树的连通子图.

参考文献

图论与枚举

I. Anderson, *A First Course in Combinatorial Mathematics*, 2nd ed., Oxford University Press, New York, 1989.

V. K. Balakrishnan, *Theory and Problems of Combinatorics*, Schaum's Outline Series, McGraw-Hill, New York, 1995.

K. Bogart, *Introductory Combinatorics*, 2nd ed., Harcourt Brace Jovanovich, San Diego, 1990.

R. Brualdi, *Introductory Combinatorics*, 4th ed., Prentice Hall, Upper Saddle River, NJ, 2004.

V. Bryant, *Aspects of Combinatorics: A Wide-Ranging Introduction*, Cambridge University Press, Cambridge, 1993.

P. Cameron, *Combinatorics: Topics, Techniques, Algorithms*, Cambridge University Press, Cambridge, 1994.

M. Erickson, *Introduction to Combinatorics*, John Wiley & Sons, New York, 1996.

R. Graham, D. Knuth, and O. Patashnik, *Concrete Mathematics: A Foundation for Computer Science*, 2nd ed., Addison-Wesley, Reading, MA, 1994.

F. Roberts and B. Tesman, *Applied Combinatorics*, 2nd ed., Prentice Hall, Upper Saddle River, NJ, 2003.

D. Stanton, R. Stanton, and D. White, *Constructive Combinatorics*, Springer-Verlag, New York, 1986.

J. VanLint and R. Wilson, *A Course in Combinatorics*, 2nd ed., Cambridge University Press, Cambridge, 2001.

图　论

C. Berge, *Graphs*, 2nd ed., Elsevier Science, New York, 1985.

N. Biggs, E. Lloyd, and R. Wilson, *Graph Theory 1736–1936*, Cambridge University Press, Cambridge, 1999.

M. Capobianco and J. Molluzzo, *Examples and Counterexamples in Graph Theory*, North Holland, New York, 1978.

C. Chartrand and L. Lesniak, *Graphs and Digraphs*, 3rd ed., Chapman and Hall, New York, 1996.

J. Gross and J. Yellen, *Graph Theory and Its Applications*, CRC Press, New York, 1998.

F. Harary, *Graph Theory*, Perseus Press, New York, 1995.

O. Ore, *Graphs and Their Uses*, Mathematics Association of America, Washington, DC, 1990.

R. Trudeau, *Introduction to Graph Theory*, Dover, New York, 1994.

D. West, *Introduction to Graph Theory*, 2nd ed., Prentice Hall, Upper Saddle River, NJ, 2001.

R. Wilson, *Introduction to Graph Theory*, 4th ed., Addison-Wesley, Reading, MA, 1997.

R. Wilson and J. Watkins, *Graphs: An Introductory Approach*, John Wiley & Sons, New York, 1990.

枚　举

M. Hall, *Combinatorial Theory*, 2nd ed., Blaisdell, Boston, 1986.

G. Polya, R. Tarjan and D. Woods, *Notes on Introductory Combinatorics*, Birkhauser, New York, 1983.

J. Riordan, *An Introduction to Combinatorial Analysis*, John Wiley & Sons, New York, 1958.

H. Ryser, *Combinatorial Mathematics*, Mathematics Association of America, Washington, DC, 1963.

R. Stanley, *Enumerative Combinatorics*, Wadsworth, Belmont, CA, 1986.

W. Whitworth, *Choice and Chance*, Hafner Press, New York, 1901.

索　　引

有限状态机, 303, 312, 315

Z